A mining research contract report
FEBRUARY 1984

ADAPTIVE-NOISE-CANCELLATION TECHNIQUES FOR THROUGH-THE-EARTH ELECTROMAGNETICS

VOLUME III — OF 3

Contract No. J0318070

Green Mountain Radio Research Company

BUREAU OF MINES
UNITED STATES DEPARTMENT OF THE INTERIOR

DISCLAIMER NOTICE

BIBLIOGRAPHIC DATA SHEET

1. Report No.	2.	3. Recipient's Accession No.	

4. Title and Subtitle
Adaptive-Noise-Cancellation Techniques for Through-the-Earth Electromagnetics, Volume III

5. Report Date
February 1984

6. Performing Organization Code
3U750

7. Author(s)
Frederick H. Raab, Ph.D.

8. Performing Organization Rept. No. TR84-1

9. Performing Organization Name and Address
Green Mountain Radio Research Company
50 Vermont Avenue
Fort Ethan Allen
Winooski, Vermont 05404

10. Project/Task/Work Unit No.

11. Contract/Grant No.
J0318070

12. Sponsoring Agency Name and Address
United States Department of the Interior, Bureau of Mines
Washington, DC 20241

13. Type of Report & Period Covered
FINAL
July 1983 - February 1984

14. Sponsoring Agency Code

15. Supplementary Notes
This report is Volume III of a three-volume set and covers work done under Phase-III of the contract.

16. Abstracts
An electromagnetic (EM) system for the location of trapped mine workers at coal-mine (300 m) depths has been developed. The use of such a system in deep mines (1-km depth) requires transmission at extremely low frequencies (ELF) and a reduction in the effective noise level beyond that which can be obtained by acceptable integration times. Automatic-noise-cancellation (ANC) techniques reduce the effective noise level by combining several received signals that contain correlated noise. Such signals can be obtained either from a remote magnetic antenna or a local electric antenna.

During Phase III of this program, real ELF noise data were acquired and used to test the ANC algorithms developed during Phase II. The ELF noise data were obtained from analog recordings made by Develco under another Bureau-of-Mines contract. Digitization was accomplished by an A/D converter that was added to a GMRR computer. Phase III tasks included preparation of software for digitization, computer-computer communication, data processing, and statistical analysis.

Useable data were obtained from eleven half-hour tapes containing noise from one axis of the primary antenna and three axes of the remote antenna. The overall average improvement in SNR was 4.1 dB. However, the average improvements for the individual X, Y, and Z axes were 4.2, 11.9, and -1.9 dB, which roughly follow the observed noise power. Nonlinear processing was not effective with this noise.

17. Key Words and Document Analysis. 17a. Descriptors

Electromagnetic direction finding
Trapped-miner location
Trapped-miner communications
Through-the-earth communications
Near-field location systems
Adaptive noise cancellation
Extremely low-frequency (ELF) noise
Adaptive clipper
Nonlinear processing

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Distribution Statement
Approved for public release, distribution unlimited.

19. Security Class (This Report)
UNCLASSIFIED

20. Security Class (This Page)
UNCLASSIFIED

21. No. of Pages
97

22. Price

FOREWARD

This report was prepared by Green Mountain Radio Research Company, Winooski, Vermont under contract J0318070. This contract was initiated under the Coal Mine Health and Safety Program. It was administered under the technical direction of the U.S. Bureau of Mines with Steve Shope acting as Technical Project Officer. Karen Hatters was the contract administrator for the Bureau of Mines. This report is a summary of the work recently completed as a part of this contract during the period July 1983 - February 1984. This report was submitted by the author in February 1984.

The manuscript and artwork were prepared by Patricia L. Scott.

CONTENTS

ILLUSTRATIONS

TABLES

CHAPTER 1. INTRODUCTION

1.1 BACKGROUND

Electromagnetic (EM) techniques and apparatus for the subsurface location of trapped mine workers have been developed during the past decade. The "first-generation" system [1 - 4] transmits signals in the 600 - 3000 Hz range to maximize the signal-to-noise ratio. Physical movement of the receiver and detection by human ear is required, and the accuracy can be degraded by irregularities in the conducting ground.

The elements of the "second-generation" EM location system [5] are shown in Figure 1-1. The subsurface beacon transmitter is similar to that of the first-generation system, but uses a lower transmitting frequency. Uplink signals are received by three three-axis magnetic-field antennas (called "search coils"). The amplitudes of the nine field components are forwarded to a computer in the electronics van. The computer processes the signals and determines the position of the subsurface transmitter. The requirements for physical movement and detection by human ear are eliminated by this system.

The deep-mine (1-km depth) application of electromagnetic (EM) trapped-miner location techniques [5] and the desire to reduce the effects of the conducting ground [6 - 9] mandate the use of lower uplink frequencies. At such frequencies (1 - 30 Hz), the ambient-noise levels [5,7,10] are considerably greater than those in the 600-to-3000-Hz band used by the first-generation EM location system. However, intrinsic-safety considerations prevent significant increases in the output of the beacon transmitter, and the link analysis [5] clearly shows that simple signal integration alone does not produce adequate signal-to-ratio in an acceptable length of time.

The primary receiving antenna is a three-axis magnetic-field sensor, and receives both the uplink signal and noise. Since the uplink signal is a quasi-static magnetic field, it decays very rapidy and has an insignificant associated electric field. In contrast, propagating atmospheric-noise fields decay very slowly and include both electric- and magnetic-field components. An electric-field antenna or a remote magnetic-field sensor therefore receives the atmospheric noise, but not the beacon signal.

Since the noises received by the primary and reference antennas are in general highly correlated, significant improvement in the signal-to-noise ratio (SNR) can be achieved by combining their outputs. Since the correlation characteristics of the antenna outputs change with time and are therefore not known *a priori*, an adaptive-noise-cancellation (ANC) algorithm is required to combine the outputs to obtain a signal with minimum noise.

A preliminary investigation of ANC techniques for through-the-earth (TTE) EM systems was conducted during Phase I of this program [11]. A thorough simulation of ELF noise, nonlinear processing, and ANC algorithms was conducted under Phase II of this program [12]. This report presents the results of Phase III, during which real ELF noise data were acquired and used to evaluate the performance of the nonlinear processing and ANC algorithms that were developed during Phase II.

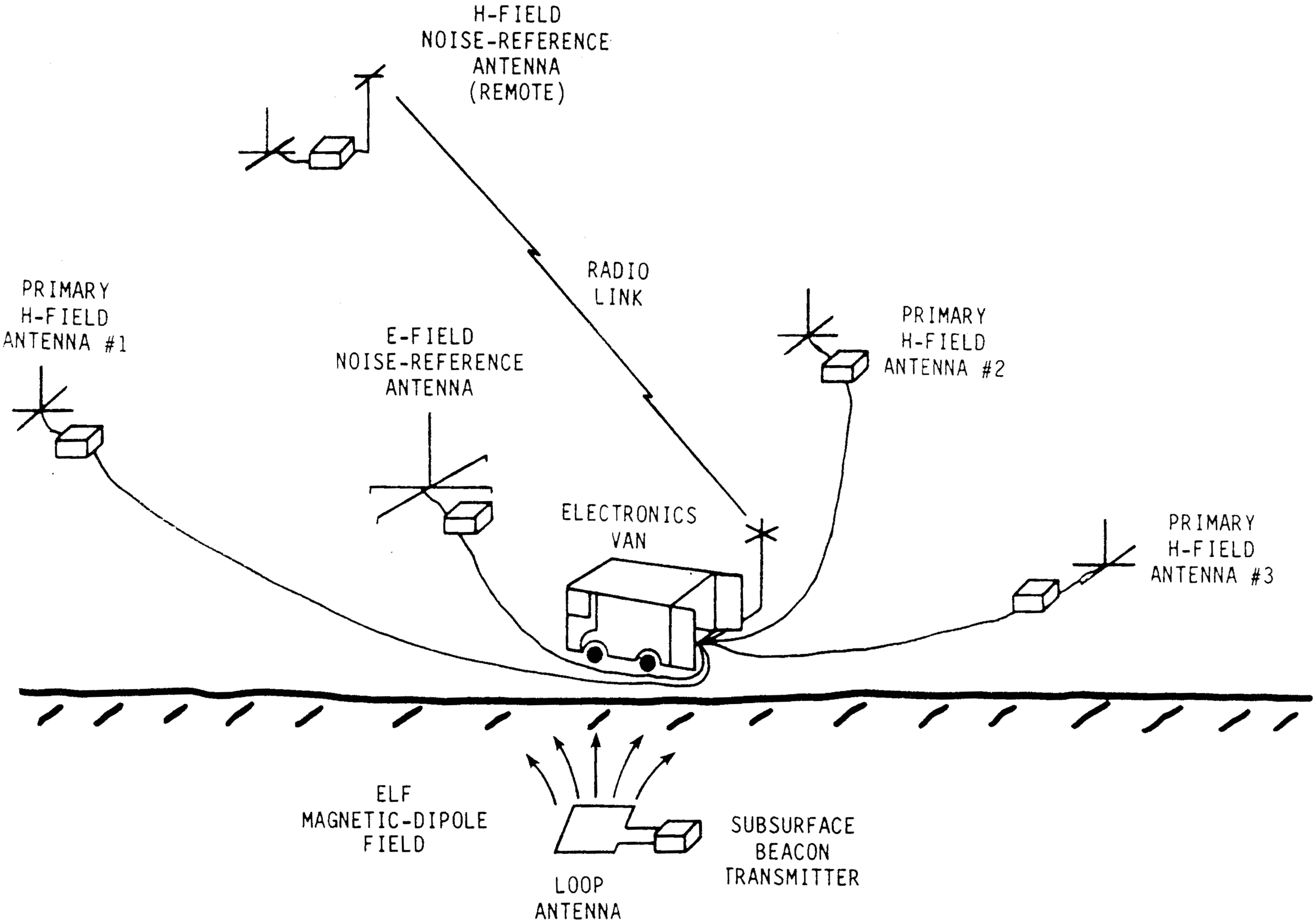


Figure 1-1. Elements of electromagnetic location system.

1.2 ANC CONCEPT

A simplified example of an automatic-noise-cancellation technique is shown in Figure 1-2. The output of the primary receiving antenna is represented by x and contains the desired signal corrupted by a significant amount of noise. The signal-to-noise ratio (SNR) in this example is about -10 dB.

Waveform y represents the output of a noise-reference antenna; in the present application, this antenna could be either a remote magnetic-field sensor or a local electric-field sensor. This received signal contains little or nothing of the desired signal, but contains noise that is highly correlated with that in the output of the primary antenna (e.g., $a \approx 1$ and $b \ll 1$).

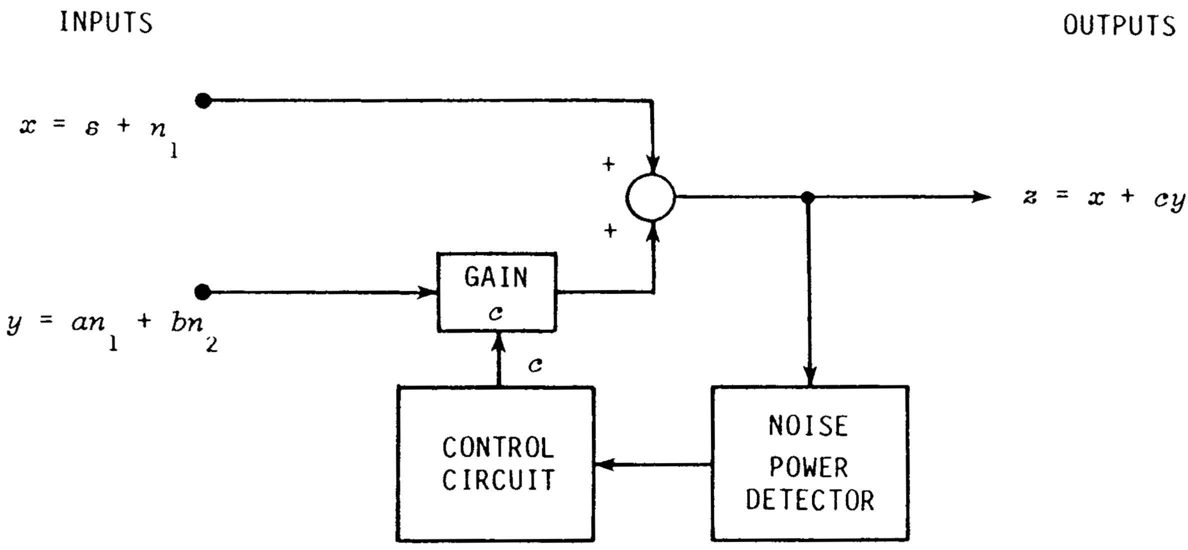
The output signal z is a linear combination of the primary and noise-reference signals. If the constant c is set properly, much of the noise is cancelled, producing the +10 dB SNR shown in Figure 1-2b.

In general, the noise statistics (hence constant c) are not known *a priori* and are time varying. It is therefore necessary to use an adaptive technique to set the value of c for maximum noise cancellation. A variety of such techniques have been developed for other applications [1].

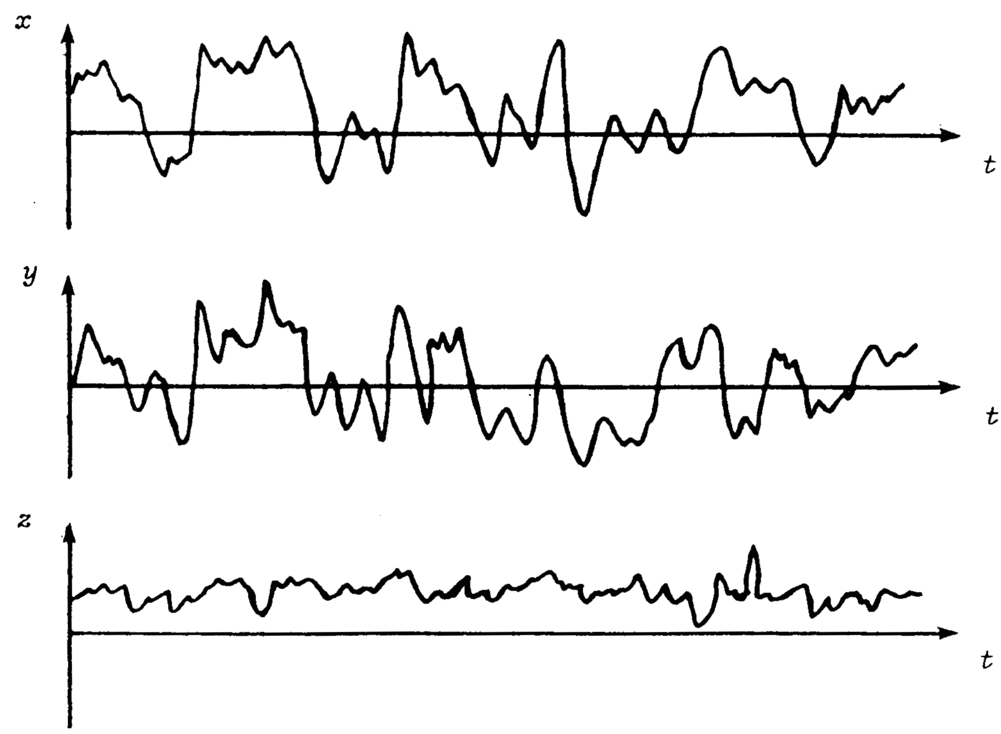
The transmitters in through-the-earth electromagnetic systems use electrically-small loop antennas and can therefore be regarded as magnetic-dipole sources. The use of extremely low excitation frequencies to reduce the effects of the conducting ground also guarantees that the electric field produced by the subsurface transmitter will be negligible within its region of operation. Consequently, the field produced is essentially a pure quasi-static magnetic dipole field having a free-space structure with an inverse-distance-cubed amplitude variation.

Atmospheric noise at extremely low frequencies is produced primarily by thunderstorms. Nearby thunderstorms produce distinct noise impulses, while the aggregate of distant thunderstorms produces a Gaussian background noise. Since much of the received noise power is contained in the impulses, it is necessary to include processing that edits (blanks) or cancels individual impulses.

Because the attenuation rates for ELF signals are very low (e.g., 0.2 dB/Mm), most of the received noise power (especially after editing) is due to distant sources. Since the wavelengths of the signals of interest are 6 Mm (3600 mi) or more, noise waveforms tend to be similar at points separated by significant distance (15 km). The primary causes of differences in the noise waveforms are local geological features (e.g., changes in ground conductivity). Nonetheless, noise waveforms with correlations from 0.9 to 0.9999 (10 to 40 dB) can be found at sites separated by as much as 15 km. (This correlation is the basis of remote-reference magnetotellurics). Since the amplitude of the desired signal decreases by 60 dB as distance is increased from 1 km to 10 km, it is apparent that a remote magnetic-field sensor can provide an essentially signal-free source of correlated noise.



(a) Block diagram



(b) Waveforms

Figure 1-2. Example adaptive noise-reduction technique.

A propagating electromagnetic signal contains both electric and magnetic-field components, which are related by the free-space impedance (377 Ω). Propagation over conducting ground produces the well-known phenomenon called "E-field tilt," which causes the electric-field to tilt in the direction of propagation. This implies that the horizontal electric-field components are strongly correlated with the opposite horizontal magnetic-field components. Since the beacon transmitter produces little, if any, electric field, it is apparent that a local electric-field antenna can also be used as a source of correlated noise. The magnetotelluric technique for geophysical prospecting is based upon this phenomenon. A noise-cancellation technique based upon electric- and magnetic-field correlation has been proposed for use with trolley phones [13].

1.3 PHASE-I RESULTS

The objective of Phase I was to conduct a preliminary investigation and feasibility assessment of ANC techniques for TTE EM systems [1]. The two principal areas of investigation were:

- ELF-noise characteristics, and
- ANC algorithms.

Phase I work also included analysis of correlation estimators.

The TTE EM location system involves reception of all three magnetic-field components and (if ANC is used) possibly all three electric-field components. The incident field includes the vertical electric and horizontal magnetic components. The Field-Lewinstein-Modestino model represents single-channel ELF noise as the sum of Gaussian and power-Rayleigh random-variables. Direction-of-arrival characteristics are accommodated by using multiple noise sources positioned to correspond to the major thunderstorm areas of the world. Horizontal-electric and vertical-magnetic field components are generated by a "geology filter" that accounts for the electromagnetic characteristics of the ground.

The limited amount of published empirical data for the frequencies of interest show correlations of both the electric- and magnetic-field components ranging from 0.9 to 0.9999. These correlation levels suggest the possibility of noise reductions from 10 to 40 dB through ANC.

Candidate ANC algorithm including direct-matrix-inversion (DMI), gradient (least-mean-square - LMS), and recursive (Kalman-filter) techniques are identified. The DMI and LMS techniques have the most potential in the TTE EM application.

1.4 PHASE-II RESULTS

The objective of Phase II was thorough comparison and evaluation by simulation of ANC techniques for the TTE EM application. The specific areas of investigation included:

- The Field-Lewinstein-Modestino (FLM) ELF-noise model,
- Direction-of-arrival (DOA) characteristics,
- The geology filter,
- Effects of reception bandwidth,
- Nonlinear-processing techniques,
- ANC algorithms, and
- The combination of nonlinear processing and ANC.

Software was developed for random-noise generation, including subroutines for uniform, Gaussian, and power-Rayleigh random-variables. The Gaussian and power-Rayleigh random variables are obtained by non-linear transformation of the uniform random variable. Proper operation is verified by moment and histogram tests.

Software for the multicomponent ELF-noise simulation model was implemented. In this model, the output from individual noise sources is obtained by summing Gaussian and power-Rayleigh random variables as specified by the FLM model. The vertical-electric-field component is obtained by summing the outputs from several such sources. The horizontal-magnetic-field components are obtained by transforming the outputs of individual sources into *X* and *Y* components and then summing the results. The unit-pulse response of a homogeneous ground is derived and used to implement a geology filter. Proper operation is verified by moment, amplitude-histogram, and DOA-histogram tests.

The FLM model characterizes ELF noise by power, impulsivity, and spikiness. However, values of the impulsivity and spikiness parameters are published for only a limited number of bandwidths. The effect of bandwidth upon these parameters was therefore analyzed, and a two-moment method for conversion was developed.

A considerable portion of the power in ELF noise is contained in discrete impulses. Nonlinear processing can therefore be used to reduce the effective received noise power. Nonlinear processors based upon editing, clipping, adaptive editing, and adaptive clipping were compared by simulation. The adaptive clipper was found to be most effective.

Narrowband DMI and LMS ANC algorithms were implemented and compared by simulation with Gaussian noise. In this application, the DMI algorithm must both estimate the noise-covariance matrix and the weighting matrix. Since errors in one estimate produce errors in the other, it is effective to apply the DMI algorithm recursively. (Two iterations are sufficient). The simulations show a gain of 0.1 produces the greatest noise reduction by the LMS algorithm. The simulations also show the DMI algorithm to produce significantly greater noise reduction than does the LMS algorithm.

The DMI algorithm was modified to operate on complex inputs that represent the narrowband outputs of the nonlinear processor (NLP). The DMI ANC and NLP algorithms were then combined; proper operation of both subsystems was verified by simulation. The combined algorithm was then tested under a variety of conditions. The benefits obtained with one algorithm alone are not additive, but the inclusion of both algorithms is nonetheless desirable. Under typical atmospheric conditions, the total noise reduction ranges from 5 to 30 dB, depending upon the local-noise level.

1.5 PHASE-III OBJECTIVES AND ORGANIZATION

The objective of Phase III is to evaluate the nonlinear-processing and adaptive-noise-cancellation algorithms with actual ELF noise. The specific areas of work include:

- Development of software for data digitization,
- Development of software for data transfer,
- Data acquisition,
- Modification of NLP/ANC software algorithms, and
- Data processing and statistical evaluation of the results.

Chapter 2 describes the data-digitization hardware and software. Digitization is accomplished by an I/O Technology 12-bit A/D converter board added to one of GMRR's North Star Horizon computers. The digitization software uses the computer's real-time clock to digitize four analog-input channels at a 100-Hz rate. A semiconductor disk emulator is used for temporary data storage to avoid problems with the track-to-track access time of the computer's floppy-disk drives.

The ELF-noise data and the data-acquisition procedure are described in Chapter 3. Data acquisition was accomplished at Develco's facility in Sunnyvale, CA. This procedure eliminated the possibility of degradation of the analog tapes by rerecording or shipment. A total of fourteen half-hour tapes were digitized, resulting in some 16 megabytes of data. Digitized data were retained by GMRR for subsequent processing and transferred to the Develco HP-1000 computer for their possible future use.

Chapter 4 discusses the data-processing software, which is based upon the simulation software developed in Phase II. Modifications to the simulation software allow operation with a single-axis primary antenna, reading noise data from a disk file, and extraction of a calibration constant. A simulation of fixed (nonadaptive) cancellation is also added.

The results of processing the data are presented in Chapter 5. Data from each analog tape were partitioned into six segments of 270-s duration. Each 270-s data set was used for one simulation run and processed with various clipping thresholds. The best results were obtained with nonlinear processing disabled. An overall average ANC gain of 4.1 dB was obtained for data from the eleven "undisturbed" tapes. The average ANC gains for the individual X, Y, and Z axes were 4.3, 11.9, and -1.9 dB, respectively.

1.6 REFERENCES

1. H. K. Sacks, "Trapped-miner location and communication systems," *Underground Mine Communications, Part 4, Section-to-Place Communications*, U.S. Bureau of Mines Information Circular 8745, 1977.

2. R. G. Olsen and A. J. Farstad, "Electromagnetic direction finding experiments for location of trapped miners," *IEEE Transactions on Geoscience Electronics*, vol. GE-11, no. 4, pp. 178 - 185, October 1973.

3. J. Durkin, "Assessment of present techniques for the location of trapped miners," *Proceedings of the Fifth WVU Conference on Coal Mine Electrotechnology*, Morgantown, WV, pp. 1-1 - 1-16, July 30 - August 1, 1980.

4. R. Gabillard, J. P. Dubus, and F. Cherperell, "Electromagnetic survey method applicable to underground quarries," *Proceedings of the Thru-the-Earth Electromagnetic Workshop*, Golden, CO, (NTIS PB 231,154), pp. 121 - 129, August 15 - 17, 1973.

5. L. H. Rorden, T. C. Moore, E. C. Fraser, and L. R. Bulduc, "Phase I development of a system concept for location of trapped miners in deep mines by an electromagnetic method," Phase I Report, contract J0199009, Develco, Inc., Sunnyvale, CA, October 1979.

6. T. W. H. Caffey, "Locating a buried magnetic dipole," *Proceedings of the Electromagnetic Guided Waves Workshop*, Boulder, CO, March 28 - 30, 1978.

7. F. H. Raab and P. K. Hansen, "Electromagnetic retransmission system for locating trapped mine workers," Final Report, contract H0188071, (PB81-215329) Polhemus Navigation Sciences, Inc., Essex Junction, VT, February 1980.

8. F. H. Raab and P. K. Hansen, "Electromagnetic technique for subsurface position determination," *Proceedings of the IEEE Industry Applications Society Annual Meeting*, Philadelphia, PA, pp. 111 - 118, October 5 - 9, 1981.

9. F. H. Raab, "Quasi-static magnetic-field technique for determining position and orientation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-19, no. 4, pp. 235 - 243, October 1981.

10. E. L. Maxwell, "Atmospheric noise from 20 Hz to 30 kHz," *Radio Science*, vol. 2, (new series), no. 6, pp. 637 - 644, June 1967.

11. F. H. Raab, "Adaptive-noise-cancellation techniques for through-the-earth electromagnetics, Volume I," Final Report GMRR-TR82-1, Green Mountain Radio Research Company, Burlington, VT, January 1982.

12. F. H. Raab, "Adaptive-noise-cancellation techniques for through-the-earth electromagnetics, Volume II," Final Report GMRR-TR83-1, Green Mountain Radio Research Company, Winooski, VT, September 1983.

13. R. L. Lagace, D. A. Curtis, J. D. Foulkes, and J. L. Rothery, "Technical service for mine communications research; Transmit antennas for portable VLF to MF wireless mine communications," Contract No. H0346045 - Task Order No. 1, Arthur D. Little, Inc., Cambridge, MA, May 1974 - May 1977.

CHAPTER 2. DIGITIZATION

ELF noise data suitable for use in evaluating the performance of ANC algorithms were collected by Develco, Inc., as part of their work on the deep-mines EM location system under Bureau-of-Mines contract J0199009. The desired noise data reside in a set of analog recordings made by a TEAC R-61 portable cassette data recorder. Four channels were recorded simultaneously, and the input bandwidth of each channel is about 30 Hz.

To evaluate performance of the various ANC algorithms on GMRR's North Star computers, it was therefore necessary to digitize the analog recordings and to store the resultant data on North Star floppy disks. This chapter describes the software and procedures developed to accomplish the data digitization and transfer.

2.1 APPROACH

The approach (Figure 2-1) selected for acquisition of the noise data required equiping a GMRR computer with an A/D converter, taking it to Develco, and digitizing the data there. This approach allowed processing of the original analog recordings and also allowed transfer of the digitized data to the Develco computer. In contrast to other possible approaches, it avoided equipment-rental fees, noise added in analog re-recording, and the possibility of damaging the original recordings during shipment.

A North Star Horizon computer is used to control data digitization and storage. This computer has a Z80 microprocessor, S-100 bus, 56 kilobytes of RAM, and two 338-kilobyte-capacity floppy-disk drives. This computer is also equipped with a 512-kilobyte SemiDisk™ (semiconductor disk emulator), and uses the CP/M-80® operating system.

Filter

The four-channel low-pass filter assembly suppresses broadband noise (0 - 1000 Hz) in the recorder output. This noise cannot be suppressed in software because of the 100-Hz sampling frequency.

The filter assembly was built by Develco engineer David Schleicher, and each channel employs a two-pole active circuit (Figure 2-2) similar to that used in Develco's deep-mine EM receiver. Each channel has unity gain at low frequency, both poles at 29-Hz, and a theoretical response of

$$F(s) = \frac{1}{(1 + s\cdot 5.49\cdot 10^{-3})^2} \quad . \tag{1}$$

Measurements of the filter response show 22 dB attenuation at 100 Hz and not more than 2° - 3° phase shift in the 0 - 30-Hz passband.

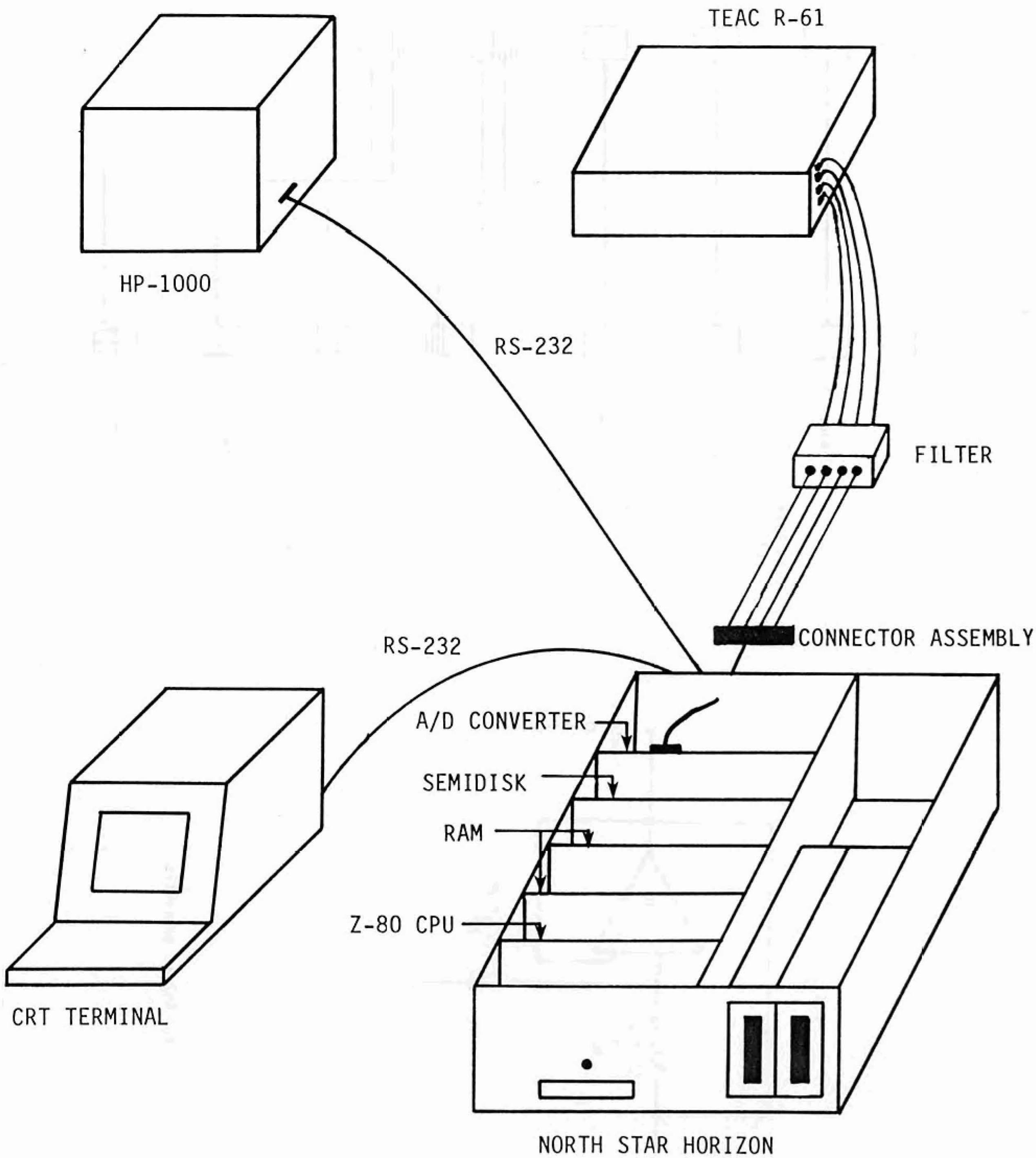


Figure 2-1. Equipment.

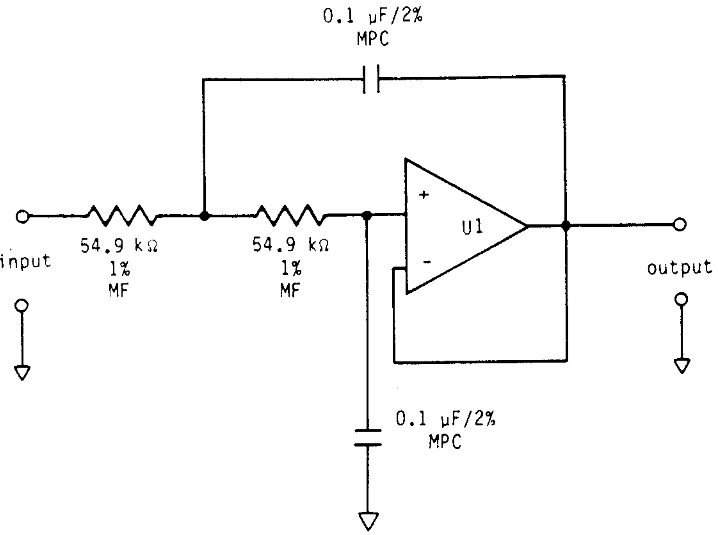


Figure 2-2. Filter schematic.

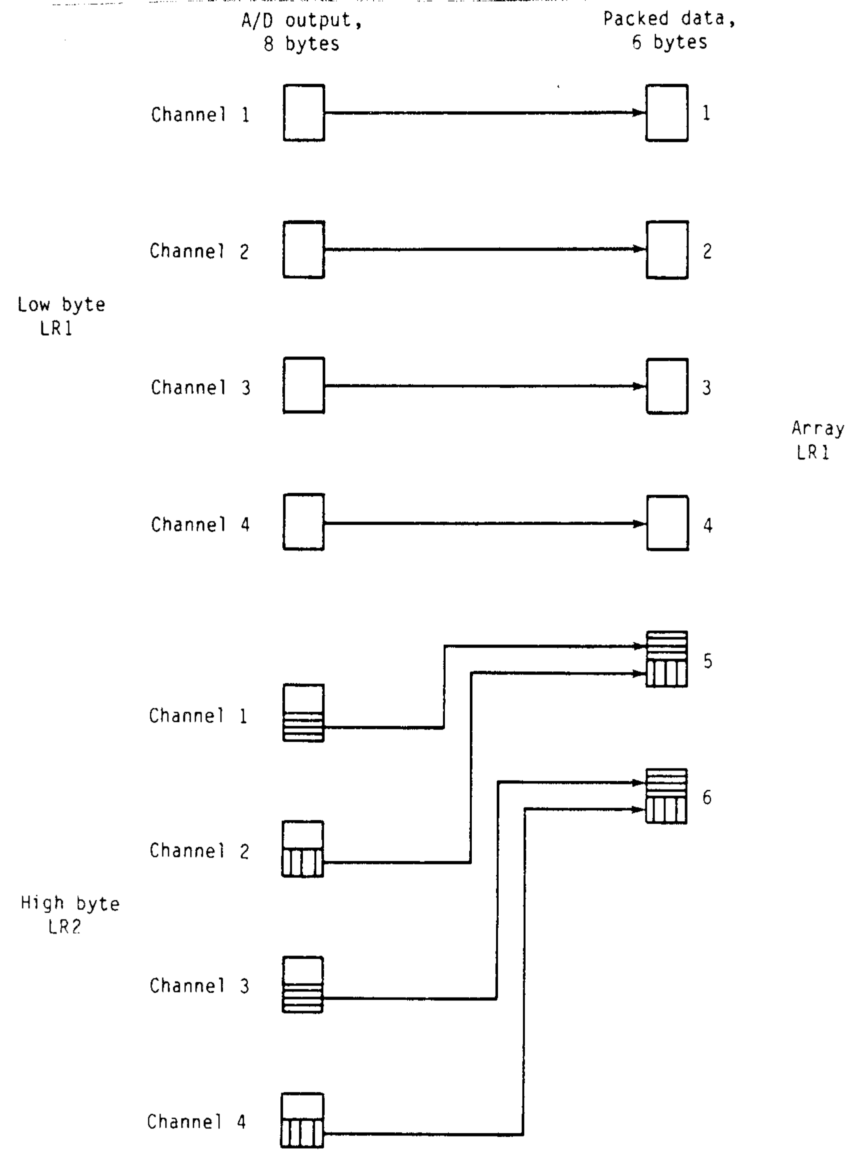


Figure 2-3. Data packing.

Digitization

Data digitization is accomplished by an I/O Technology S-100 A/D/A converter board installed in the North Star computer. This board provides eight single-ended channels and has a built-in auto-zero capability for elimination of dc-offset error.

The board requires 12 μs per sample when sampling one channel with a constant gain and 24 μs per sample when changing channels and/or gains; its speed is therefore more than adequate for the desired 100-Hz sampling frequency. Its 12-bit accuracy corresponds to a 72-dB dynamic range, which should also be more than adequate, since the specified SNR of the TEAC R-61 recorder is only 50 dB. The converter has gain settings of 0.5, 2.0, 8.0, and 32.0 and a maximum measurement voltage (after application of gain) of ±5 V; these levels are compatable with the ±1.5-V maximum output of the TEAC R-61. The 10-MΩ input impedance of the A/D/A board should also be compatable with the 600-Ω output of the TEAC R-61.

A 100-Hz sampling frequency was selected to ensure that no information in the 30-Hz-bandwidth analog signals is lost in the sampling process. Since the 12-bit output from the A/D converter can be packed into 1.5 bytes, simultaneous digitization of four channels with a 100-Hz sampling frequency produces data at the rate of 600 bytes/s. A forty-minute analog tape therefore produces a total 1,440,000 bytes.

Storage

The North Star floppy-disk drive is capable of storing data at the rate of 695 bytes/s on the average. However, the specified track-to-track access time of 40 ms is four times the 10-ms sampling interval. It is therefore impractical to store A/D output directly on floppy disk.

Data can be written to the SemiDisk by the operating system at an average rate of 3360 bytes/s. However, data are transferred to a disk only in 128-byte records, and 12 ms are consumed in each such transfer.

It is therefore necessary to transfer data to the SemiDisk on a byte-by-byte basis using I/O calls. Tests show that the average time required to transfer one byte via this mechanism is 0.09 ms. The worst-case time required to transfer one byte is 0.2 ms, and occurs when a new sector and/or track must be specified through additional I/O calls. Transfer of the six bytes generated by one set of four samples therefore consumes at the most only 1.2 ms. Since the time required is only a small fraction of the 10-ms sampling period, this mode of access is satisfactory.

2.2 A/D-INTERFACE SUBROUTINES

The set of subroutines described in this section is used to control the I/O Technology A/D/A converter board [1] and to interpret its output. The subroutines are listed in Appendix A.

Access

The converter board can be jumpered to allow control by either memory or I/O references. The required number of addresses (128) is rather large for the 8-bit I/O address space of 8080-type microprocessors. Since the North Star Horizon disk controller uses memory addresses in the range of E8ØØH to EB99H, memory from ECØØH to FFFFH is not generally utilized. The memory block from ECØØH to EFFFH can therefore be disabled, allowing the use of these addresses for control of the A/D converter. The use of memory control results in slightly faster access than is possible with I/O control.

Control Bytes

Subroutine ADSET assembles the two bytes required to order an A/D conversion, and computes the addresses for each. The "upper byte" contains the bias used for offset-voltage correction. The "lower byte" includes the channel number, gain, and whether single-ended or differential measurements are to be made.

Conversion

Subroutine ADCNV orders an A/D conversion and returns the result. To allow adequate settling time after changing channel or gain settings, two conversions are ordered and only the output from the second is actually returned. Provision is included for waiting for the conversion-complete signal. However, in practice the various computer instructions that must be carried between ordering the conversion and reading out the result take up more time than the conversion. (The North Star Horizon running a program written in Microsoft FORTRAN-80 requires 26 μs to do a PEEK and 9 μs to do an IF.)

Interpretation of Output

Subroutine ADVOLT converts the two-byte result of a conversion into a floating-point voltage. Eight bits from the lower byte and three bits from the upper byte are assembled into an integer. The integer is then converted into a floating-point variable and scaled. The sign is determined from bit four of the upper byte and the voltage is modified if necessary. Finally, the voltage is scaled to include the converter gain.

Auto Zero

Subroutine ADBIAS automatically finds the bias byte that brings the output voltage closest to zero. A binary-search procedure with a total of ten iterations is used. The analog-input lines must be grounded for the auto-zero procedure to be effective.

Gain Selection

Subroutine ADBIAS queries the user for the maximum anticipated input voltage and sets the gain to the maximum value that can be used without saturating the converter.

2.3 SEMIDISK-INTERFACE SUBROUTINES

The SemiDisk™ is a 512-kilobyte random-access memory that responds to I/O calls rather than memory references, thus allowing its use in an 8080-type environment with 16-bit memory addressing [2]. The SemiDisk is accessed through four I/O addresses, which are used to specify data byte, byte number (0 - 127), sector number (0 - 15), and track number (0 - 255). The set of five subroutines listed in Appendix A and explained below provide convenient, rapid access to the SemiDisk.

Initialization

The SemiDisk is initialized by subroutine SDINIT. SDINIT sets up the I/O addresses (8ØH - 83H in this system), sets the byte, sector, and track counters to zero, and (optionally) zeros all SemiDisk memory.

Read and Write

Subroutine SDREAD reads one byte of data from the SemiDisk at the location currently specified by the byte, sector, and track counters. Subroutine SDWRIT similarly writes one data byte to the SemiDisk.

Set and Increment

Subroutine SDSET is used to write desired byte, sector, and track numbers to the SemiDisk. Subroutine SDINCR increments these counters in an appropriate manner after a byte is read or written.

2.4 OPERATIONAL PROGRAMS

Operational use and checking of the A/D system is accomplished by a set of three programs:

- "AD" digitizes four analog inputs and stores the results in the SemiDisk.
- "SF" transfers data from the SemiDisk to floppy disk at the end of a digitizing session.
- "FD" decodes data from a floppy disk and displays the corresponding set of voltages for each sample set.

Program listings are given in Appendix A. Disk backup is accomplished by the standard CP/M utilty "COPY".

Digitization

Upon start-up, the "AD" program requires the user to respond to five questions regarding

- Physical-channel selection,
- Maximum input voltage,
- Bias-correction method,
- Duration of processing, and
- Sampling interval.

Any set of four of the eight A/D-converter input channels can be selected for digitization. The gain is set to the maximum value consistant with the maximum anticipated input voltage. Bias correction (auto-zero) can be accomplished by assumption, measurement, or reading correction bytes from a disk file. The values of the correction bytes are displayed by the program, and must be placed in the first line of the disk file in FORTRAN 4I5 format if that option is to be used in subsequent runs.

The duration of the processing is specified by the number of 100-sample data sets (four-channels) to be processed. For the desired 100-Hz sampling frequency, this number is the number of seconds of data to be processed. The sampling frequency is set by specifiying the number of 3.3-ms intervals in one sampling interval. Setting this number to 3 produces a 10-ms sampling interval, hence the desired 100-Hz sampling frequency.

Subroutine PACK compresses the eight-byte converter output for a four-channel data set into six bytes for storage. As shown in Figure 2-3, the four lower bytes (in which all eight bits are utilized) are stored without change. However, the low-order nibbles from four upper bytes are packed into two bytes.

Timing is controlled by the real-time clock in the North Star Horizon, which is jumpered to generate 3.3280-ms intervals; three such intervals and the related instructions produce the desired 10-ms sampling interval (Figure 2-4). The total time required for sampling, packing, and storing the data is 2.0 ms (1.4 ms for sampling, 0.6 ms for packing and storing). To allow for any possible variation in time requirements, sampling is undertaken during the first 3.3-ms interval and packing/storing is undertaken during the second 3.3-ms interval. One or more 3.3-ms intervals are added to obtain a sampling interval of 10-ms or longer duration. Tests show the resultant 100-Hz sampling frequency to be accurate to within ± 0.001 Hz.

Each time it is used, the program requests from the operator an ASCII data-identification label, whose length can be up to 64 characters. The first 128 bytes of data stored on the SemiDisk (hence subsequently on floppy disk) include this label and other parameters such as gain, number of samples, and duration of the sampling interval. The format of this data block is shown in Table 2-1.

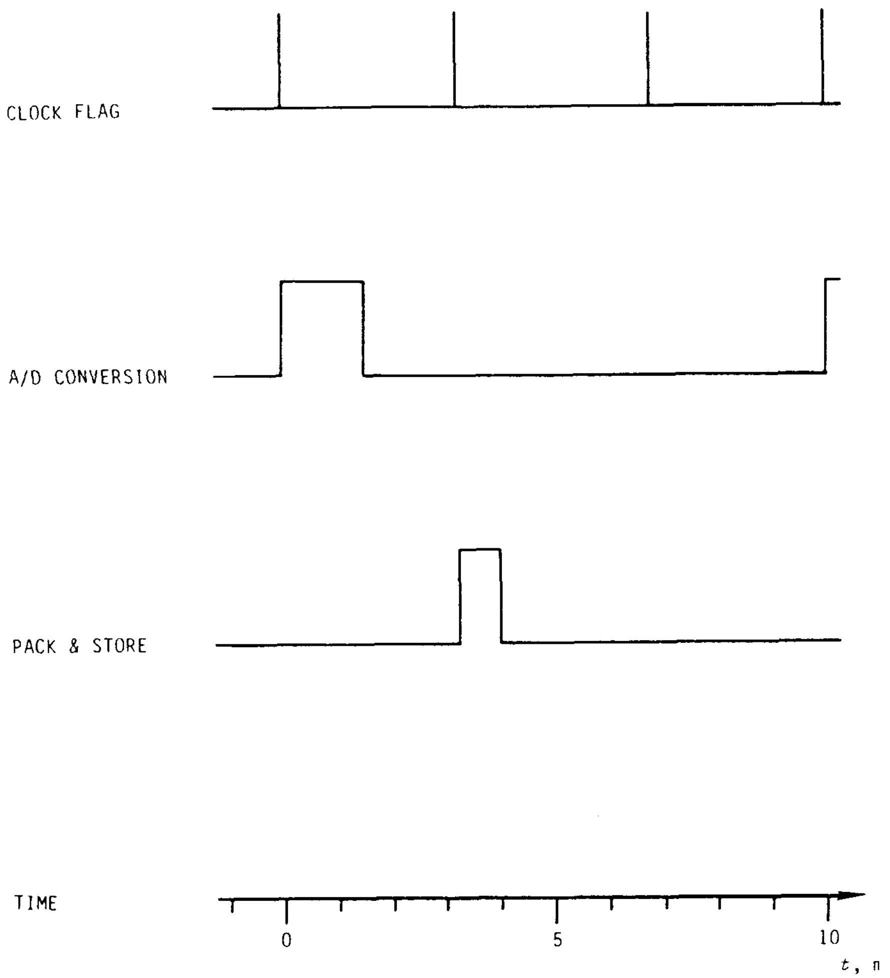


Figure 2-4. Timing.

BYTE NUMBERS	USE
1 - 64	Data-identification label - ASCII text
65	Gain code (0: 0.5, 1: 2.0, 2: 8.0, 3: 32.0)
66 - 67	Number of 100-sample sets
68 - 69	Number of 3.3-ms intervals per sample
70 - 128	Unused, set to zero

Table 2-1. Data-identification block.

Transfer to Floppy Disk

Program "SF" transfers data from the SemiDisk to a floppy disk. The program requires the operator to specify the number of 100-sample sets and the filename of the floppy-disk file. Data are transferred to the disk in packed form to maximize its storage capability. The operating system imposes a read-after-write check to ensure data integrity.

Data Decoding

The data stored on a floppy disk by program "SF" can be decoded and displayed by program "FD". The data-identification block is decoded first, and the label, gain, number of 100-sample sets, number of 3.3-ms samples, and sampling frequency are displayed on the console. The program then begins unpacking and decoding the data. Each set of four voltages is displayed along with the sample-set and sample numbers.

2.5 OPERATIONAL PROCEDURES

The maximum length of a digitizing session is limited by the 338-kilobyte (346,112-byte) capacity of the floppy disk. For a sampling frequency of 100 Hz, the maximum session duration is 576 s (9.61 minutes). Prior to beginning to digitize a new tape, the operator must determine the duration of the recording and break it into a suitable set of two or three intervals. Convenient session lengths are 540 s (9.0 minutes) and 570 s (9.5 minutes).

To digitize the data, the operator loads the program, answers its questions, and readies the recorder. The operator then starts the recorder and immediately thereafter presses "RETURN" on the console to start digitizing the data. As the end of the session approaches, the operator must be ready to stop the recorder upon the bell signal from the console.

Data must be transferred to the floppy disk before any additional data are digitized. The operator must be sure to supply "FD" with the correct number of data sets. Transfer of data to the floppy disk is only slightly faster than digitization (e.g., 555 s vs. 575 s). If a disk-write error occurs, the transfer must be repeated.

The operator should then make a duplicate (backup) of the disk using the CP/M "COPY" program. A disk copy takes about two minutes.

After all data are digitized and stored in duplicate on floppy disks, they can be transferred to the Develco computer. This transfer is accomplished via RS-232 ports and the communication program described in Appendix B.

2.6 REFERENCES

1. A/D/A Analog to Digital and Digital to Analog Converter Board User Manual, P/N 52748-900, Revision 5, 3/83, I/O Technology, Canyon Country, CA, 1982.

2. The 4931 512K S-100 SemiDisk and The 4931-1 1Mbyte S-100 SemiDisk Manual, SemiDisk Systems, Beaverton, OR, 1982.

CHAPTER 3. ACQUISITION

ELF noise data were acquired during a visit by the author to the Develco, Inc. plant in Sunnyvale, California. The principal objectives were:

- To digitize the analog noise recordings,
- To store the data on North Star diskettes for use by GMRR, and
- To transfer the data to an HP-1000 for use by Develco.

The digitization hardware, software, and procedure are described in Chapter 2. This chapter describes the noise data and the software for their transfer to the HP-1000.

3.1 NOISE DATA

Fourteen analog tapes (Table 3-1) containing two-site noise data were digitized. The first three channels of these tapes correspond to the X, Y, and Z axes of the remote-antenna. The fourth channel is either the X, Y, or Z axis (Table 3-1) of the primary antenna.

Each tape contains approximately thirty minutes of noise data, and is digitized in three 575-s (9 min, 35 s) segments and one 120-s (2 min) segment. All tapes begin with a sequence of ± 1-V calibration signals. Digitization was accomplished with the converter gain set at 2.0 to allow for the maximum ± 1.5-V output of the recorder.

The primary and remote-reference antennas were located at Stone Canyon and Bear Valley, respectively. These sites are near Hollister, CA, and are separated by approximately eight kilometers (Figure 3-1). General recording and digitization parameters are listed in Table 3-2

The fifteenth data set shown in Table 3-1 contains three examples of system noise. Set # 15-1 is a digitization of a recording of the "0-V" calibration signal of the recorder. Set # 15-2 is a digitization of a recording made with no calibration signal and the recorder inputs shorted. Set # 15-3 is a digitization of the intrinsic converter noise with the converter inputs shorted. All three system-noise data sets contain 575-s of data and all four channels were subjected to the same conditions.

The voltage output from the recorder can be converted into received magnetic-field intensity in Amperes per meter through a two-step process. First, the voltage must be divided by a factor of approximately 0.29, which corresponds to the measured calibration-signal voltage. The result is then multiplied by $12 \cdot 10^5$ to obtain the magnetic field at the antenna.

Develco engineers anticipated typical noise levels of -120 dB relative to 1 A/m in a 1-Hz bandwidth. However, the observed noise levels were in the considerably less; e.g., -141 dB, -137 dB, and -147 dB for the X, Y, and Z axes, respectively. Tests using two nearby antennas to cancel ambient noise show a system-noise level of -153 dB.

TAPE ID	PRIMARY AXIS	PRIMARY DISK	BACKUP DISK	SECONDS RECD	FILE-NAME	TAPE COUNTER	CALIBRATION SIGNAL
1	Y	Jan. 14, 1983, 16:45. Good weather, some disturbance.					
		527	479	575	ND1-1		Y
		528	480	575	ND1-2		N
		529	481	575	ND1-3		N
		530	482	120	ND1-4		N
2	Z	Jan. 14, 1983, 18:00. Good weather.					
		531	483	575	ND2-1	000 - 219	Y
		532	484	575	ND2-2	219 - 376	N
		533	485	575	ND2-3	376 - 507	N
		534	486	120	ND2-4	507 - end	N
3	Y	Jan. 5, 1983, 16:00.					
		535	487	575	ND3-1	000 - 196	Y
		536	488	575	ND3-2	196 - 351	N
		537	489	575	ND3-3	351 - 482	N
		538	490	120	ND3-4	482 - end	N
4	X	Jan. 11, 1983, 05:30. Good weather.					
		539	491	575	ND4-1	000 - 220	Y
		540	492	575	ND4-2	220 - 378	M
		541	493	575	ND4-3	378 - 509	N
		542	494	120	ND4-4	509 - end	N
5	Y	Jan. 11, 1983, 06:20. Good weather.					
		543	495	575	ND5-1	000 - 216	Y
		544	496	575	ND5-2	216 - 373	N
		545	497	575	ND5-3	373 - 503	N
		546	498	120	ND5-4	503 - end	N
6	Z	Jan. 5, 1983, 18:00.					
		547	499	575	ND6-1	000 - 219	Y
		548	500	575	ND6-2	219 - 377	N
		549	501	575	ND6-3	377 - 508	N
		550	502	120	ND6-4	508 - 535	N
7	Z	Jan. 11, 1983, 07:30. Good weather.					
		551	503	575	ND7-1	000 - 218	Y
		552	504	575	ND7-2	218 - 375	N
		553	505	575	ND7-3	375 - 505	N
		554	506	120	ND7-4	505 - 533	N
8	X	Jan. 7, 1983, 09:40. Good weather.					
		555	507	575	ND8-1	000 - 219	Y
		556	508	575	ND8-2	219 - 378	N
		557	509	575	ND8-3	378 - 510	N
		558	510	120	ND8-4	510 - 536	N
9	X	Jan. 5, 1983, 11:10.					
		559	511	575	ND9-1	000 - 220	Y
		560	512	575	ND9-2	220 - 378	N
		561	513	575	ND9-3	378 - 510	N
		562	514	120	ND9-4	510 - 536	N
10	Z	Jan. 7, 1983, 11:20. Good weather.					
		563	515	575	ND10-1	000 - 219	Y
		564	516	575	ND10-2	219 - 377	N
		565	517	575	ND10-3	377 - 507	N
		566	518	120	ND10-4	507 - end	N

Table 3-1. List of tapes, files, and conditions

TAPE ID	PRIMARY AXIS	PRIMARY DISK	BACKUP DISK	SECONDS RECD	FILE-NAME	TAPE COUNTER	CALIBRATION SIGNAL
11	X	Jan. 6, 1983, 19:15.					
		567	519	575	ND11-1	000 - 220	Y
		568	520	575	ND11-2	220 - 378	N
		569	521	575	ND11-3	378 - 509	N
		570	522	120	ND11-4	509 - end	N
12	Y	Jan. 12, 1983, 19:30. Good weather.					
		571	523	575	ND12-1	000 - 219	Y
		572	524	575	ND12-2	219 - 376	N
		573	525	575	ND12-3	376 - 506	N
		574	526	120	ND12-4	506 - 533	N
13	X	Jan. 14, 1983, 14:00. Good weather, disturbed data.					
		575	587	575	ND13-1	000 - 219	Y
		576	588	575	ND13-2	219 - 376	N
		577	589	575	ND13-3	376 - 507	N
		578	590	120	ND13-4	507 - end	N
14	Y	Jan. 7, 1983, 10:30. Good weather.					
		579	591	575	ND14-1	000 - 219	Y
		580	592	575	ND14-2	219 - 376	N
		581	593	575	ND14-3	376 - 507	N
		582	594	120	ND14-4	507 - end	N
15	Noise diagnostic measurements - same conditions all four channels						
	N/A	583	595	575	ND15-1	Calib "0-V"	Y
	N/A	584	596	575	ND15-2	Rcdr, shorted	Y
	N/A	585	597	575	ND15-3	A/D, shorted	N

Table 3-1. List of tapes, files, and conditions (continued).

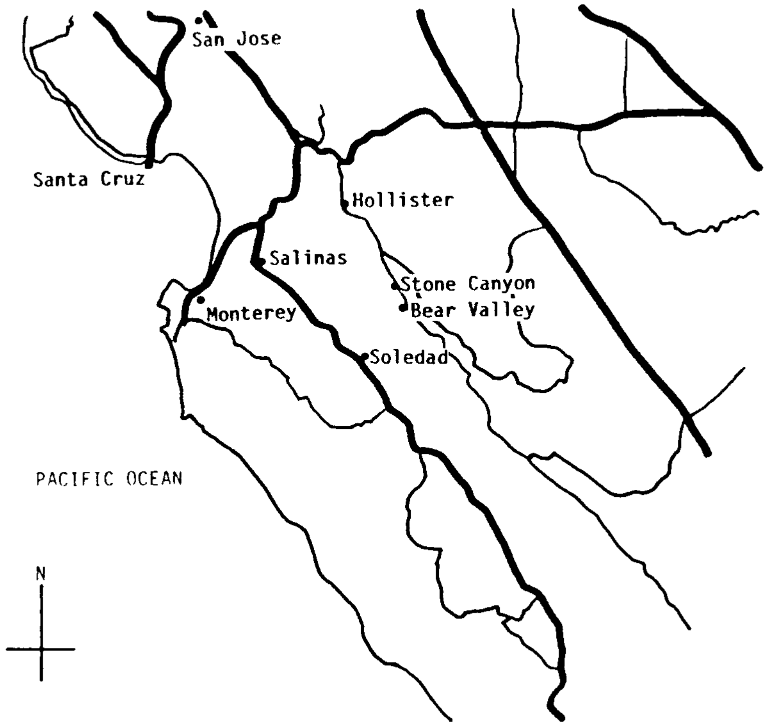


Figure 3-1. Antenna sites.

A/D CHANNELS		ANTENNA SITES	
# 1:	Remote antenna, X axis	Remote:	Bear Valley, CA
# 2:	Remote antenna, Y axis	Primary:	Stone Canyon, CA
# 3:	Remote antenna, Z axis		
# 4:	Primary antenna, specified axis		

GAIN

Gain = 2.0

Table 3-2. General recording/digitizing parameters.

3.2 STORAGE AND TRANSFER

The digitized data from each analog tape were stored on a set of four floppy disks. The initial portion of each data subset was validated by inspection with the file-display program (FD). A backup copy of each disk was made. A total of 16.4 megabytes of digitized noise data was produced.

The data were transferred to Develco's HP-1000 computer via an RS-232 interface. While this proved to be rather slow, the total time consumed was less than that required to implement a high-speed parallel interface.

Communication Protocol

A binary-block protocol was used to maximize the data-transfer rate. This protocol is based upon the transmission of a block of 128 8-bit data bytes, followed by a single parity-check byte. The check byte is obtained by exclusive-ORing the 128 data bytes, and detects all single errors. The receiving computer compares the received check byte with one computed from the received data bytes and issues a positive or negative acknowledgement. The transmission is repeated up to two additional times when a negative acknowledgement is received.

To accommodate the requirements of the HP-1000 (receiving computer), the protocol, as seen by the transmitting (North Star Horizon) computer, consists of the following steps:

- Wait for D_1 from HP-1000.
- Send one character:

  S_X to start data block

  S_B to end file

  E_T to end program.
- Send 128 data bytes.
- Send check byte.
- Await E_N, send A_K.
- Await E_N, send A_K.
- Receive

  A_K for successful reception.

  N_K for unsuccessful reception.

- Await E_N, send A_K.
- Await E_N, send A_K.
- Await E_N, send A_K.

Program Operation

The transmitter program is listed in Appendix B.

The transmitter program must be started first. Before data can be transferred, the operator must supply I/O-port parameters in response to the questions asked by the program. The I/O-port information is given in the CP/M BIOS listing and the computer manual.

During operation, the program displays the number of the disk record that is currently being transferred.

The program can be permanently configured for a particular computer by using the CP/M-80 "SAVE" command. The required steps are:

- Find the program size (in 128-byte disk records) using "STAT".
- Divide the number of disk records by two to obtain the number of 256-byte memory pages.
- Run the program and supply the desired I/O parameters.
- Interrupt the program by typing "E_X" (control-C, ASCII #3) in response to "FILENAME = ".
- Use "SAVE" to save the memory image in a disk file.

A program that has been "SAVEd" will never again request I/O information from the operator.

CHAPTER 4. PROCESSING

Software for testing the wideband NLP/ANC (nonlinear-processing / adaptive-noise-cancellation) algorithm with digitized ELF noise was developed through modification of the simulation program [1, Chapter 7] developed in Phase-II of this program. The specific steps in this effort were:

- Modification of the NLP/ANC simulation program for a single-channel primary antenna,
- Inclusion of an optional simulation of quantization by the A/D converter,
- Further modification of the single-channel simulation program to output simulated digitized noise to a disk file,
- Further modification of the single-channel simulation program to obtain its noise inputs from a disk file, and
- Modification of the disk-file processing program to add a fixed (nonadaptive) noise-cancellation processor.

In addition to the development of the processing program, cursory examinations of the differences between channels, effects of quantization, and effects of cancellation method were made.

4.1 SINGLE-CHANNEL SIMULATOR

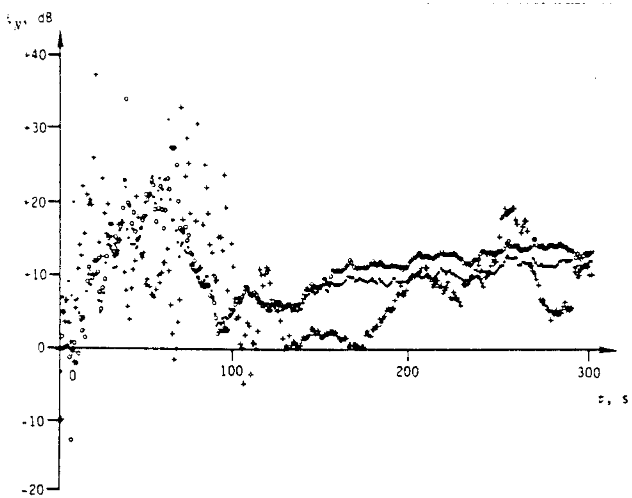
The digitized ELF noise (Chapter 3) includes all three axes of the remote-reference antenna, but only one axis is the primary antenna. The wideband NLP/ANC simulation program [1, Chapter 7] was therefore modified for operation with a selectable (X, Y, or Z) single-axis primary-antenna input.

A limited number of simulations were conducted using parameters analogous to those of Figure 7-24 of [1]. However, the signal and noise amplitudes were scaled to resemble those of the digitized ELF noise. The results are shown in Figures 4-1 and 4-2 and Table 4-1. It is apparent that the behaviors of estimators for the X and Z axes are similar. However, any correlation between the behavior of the Y-axis estimate and those of the other two axes is rather vague.

4.2 QUANTIZATION

The single-channel NLP/ANC-simulation program includes optional simulation of quantization and limiting by the A/D converter (Chapter 2). The results are shown in Table 4-1 and Figures 4-3 - 4-6.

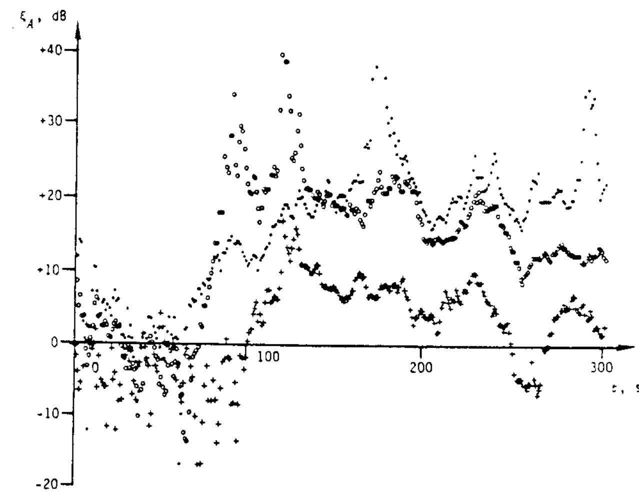
The convergence of unquantized and quantized simulations are, in general, similar. Differences in the processing gains are generally no more than



. X   + Y   o Z

$P_1 = P_3 = 0.05$, $P_2 = P_4 = 0.0$, $P_L = 0.001$
$Y_1 = Y_3 = 2.6$, $\alpha_1 = \alpha_3 = 0.28$
$Y_{xx} = 0.6$, $Y_{xy} = 0.4$, $Y_{zz} = 0.0$
$s = 0.08115$
Magnetic reference antenna
$f_{samp} = 100$ Hz, $B = 1$ Hz
Seeds: 16379, 16656

Figure 4-1. Effect of channel upon NLP improvement.



. X   + Y   o Z

$P_1 = P_3 = 0.05$, $P_2 = P_4 = 0.0$, $P_L = 0.001$
$Y_1 = Y_3 = 2.6$, $\alpha_1 = \alpha_3 = 0.28$
$Y_{xx} = 0.6$, $Y_{xy} = 0.4$, $Y_{zz} = 0.0$
$s = 0.08115$
Magnetic reference antenna
$f_{samp} = 100$ Hz, $B = 1$ Hz
Seeds: 16379, 16656

Figure 4-2. Effect of channel upon ANC improvement.

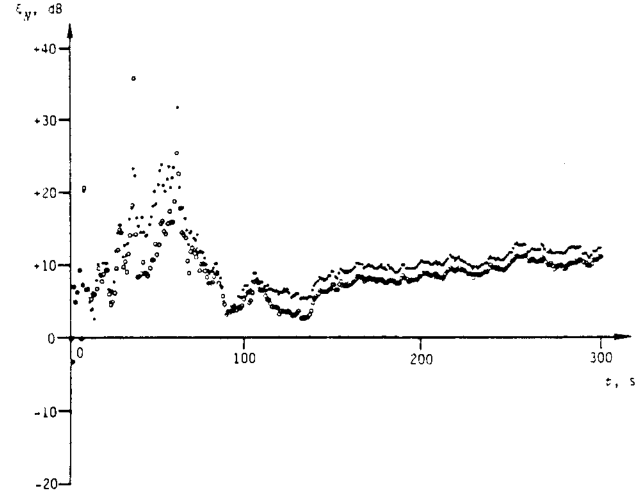
Test No.	Axis	NL	U/Q	FINAL ϵ_L	FINAL ϵ_N	FINAL ϵ_A	FINAL ξ_N	FINAL ξ_A	FINAL ξ_T	AVERAGE ξ_N	AVERAGE ξ_A	AVERAGE ξ_T
1	X	G	U	.1435E-02	.5450E-03	.3516E-03	8.4	3.8	12.2	6.0	14.2	20.1
2	X	G	Q	.1433E-02	.5417E-03	.3559E-03	8.4	3.6	12.1	6.0	14.2	20.1
D							0.0	-0.2	-0.2	0.0	0.0	0.0
3	X	H	U	.4259E-02	.8924E-03	.1148E-03	13.6	17.8	31.4	12.7	17.6	30.3
4	X	H	Q	.3199E-02	.7518E-03	.8542E-04	12.6	18.9	31.5	11.2	18.6	29.8
D							-1.0	1.1	0.1	-1.5	1.0	-0.5
5	Y	M	U	.1256E-02	.3958E-03	.2841E-03	10.0	2.9	12.9	8.2	4.5	12.7
6	Y	M	Q	.1170E-02	.4120E-03	.2582E-03	9.1	4.1	13.1	10.8	6.2	17.0
D							-0.9	1.2	0.2	2.6	1.7	4.3
7	X	M	U	.4128E-02	.9894E-03	.7286E-04	12.4	22.7	35.1	10.8	23.5	34.4
8	X	M	Q	.3201E-02	.8774E-03	.1329E-03	11.2	16.4	27.6	9.3	20.8	30.1
D							-1.2	-6.3	-7.5	-1.5	2.7	3.4
9	Z	M	U	.2798E-02	.5695E-03	.1378E-03	13.8	12.3	26.2	13.0	16.6	29.5
10	Z	M	Q	.2391E-02	.4664E-03	.1221E-03	14.2	11.6	25.8	13.2	17.9	31.1
D							0.4	-0.7	-0.4	0.2	1.3	1.6
11	X	L	U	.2957E-02	.6564E-03	.4182E-04	13.1	23.9	37.0	10.1	20.0	30.1
12	X	L	Q	.2969E-02	.6685E-03	.4342E-04	12.9	23.7	36.7	10.0	20.1	30.1
D							-0.2	-0.2	-0.3	-0.1	0.1	0.0

NL: Noise Level

U: Unquantized Q: Quantized

All gains in decibels.

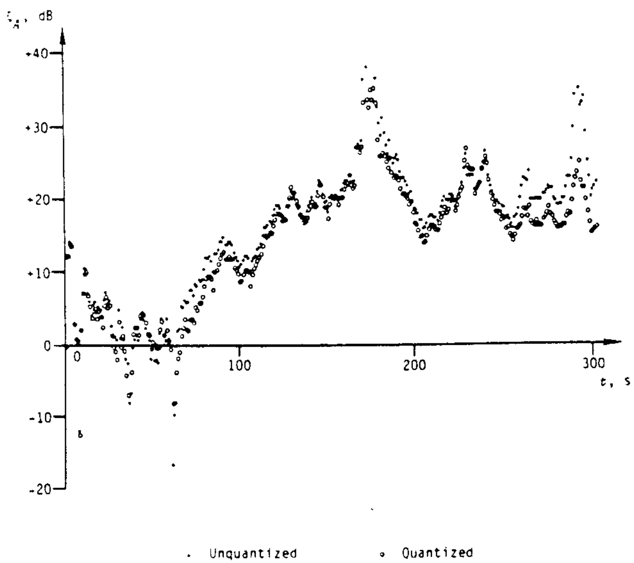
Table 4-1. Simulation results.



• Unquantized ∘ Quantized

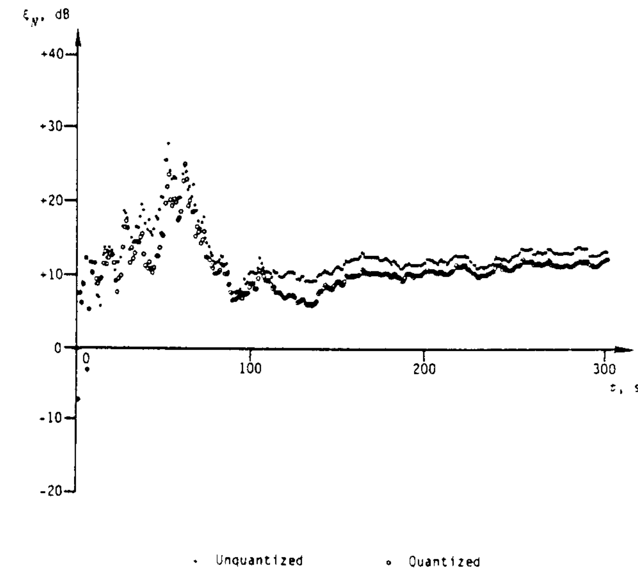
$P_1 = P_3 = 0.05$, $P_2 = P_4 = 0.0$, $P_L = 0.001$
$\Upsilon_1 = \Upsilon_3 = 2.6$, $a_1 = a_3 = 0.28$
$Y_{xx} = 0.6$, $Y_{xy} = 0.4$, $Y_{xz} = 0.0$
$a = 0.08115$
Magnetic reference antenna
$f_{samp} = 100$ Hz, $B = 1$ Hz
Seeds: 16379, 16656

Figure 4-3. Effect of quantization upon NLP improvement, X axis, moderate-level noise.



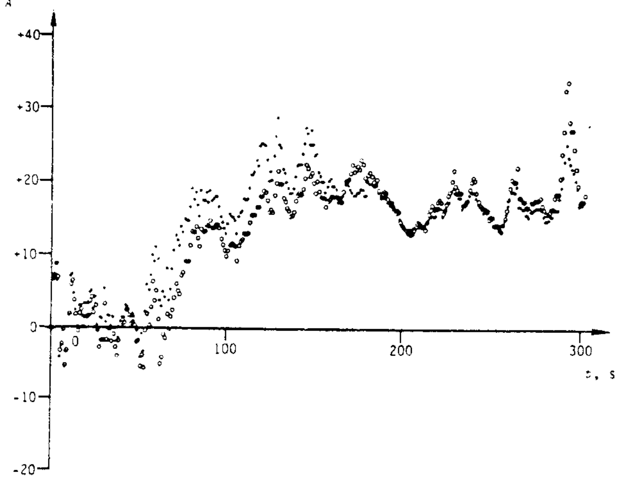
$P_1 = P_3 = 0.05$, $P_2 = P_4 = 0.0$, $P_L = 0.001$
$\gamma_1 = \gamma_3 = 2.6$, $\alpha_1 = \alpha_3 = 0.28$
$\Upsilon_{xx} = 0.6$, $\Upsilon_{xy} = 0.4$, $\Upsilon_{xz} = 0.0$
$s = 0.08115$
Magnetic reference antenna
$f_{samp} = 100$ Hz, $B = 1$ Hz
Seeds: 16379, 16656

Figure 4-4. Effect of quantization upon ANC improvement, X axis, moderate-level noise.



$P_1 = P_3 = 0.05$, $P_2 = P_4 = 0.0$, $P_L = 0.001$
$\gamma_1 = \gamma_3 = 2.6$, $\alpha_1 = \alpha_3 = 0.28$
$\Upsilon_{xx} = 0.6$, $\Upsilon_{xy} = 0.4$, $\Upsilon_{xz} = 0.0$
$s = 0.08115$
Magnetic reference antenna
$f_{samp} = 100$ Hz, $B = 1$ Hz
Seeds: 16379, 16656

Figure 4-5. Effect of quantization upon NLP improvement, X axis, high-level noise.



· Unquantized ∘ Quantized

$P_1 = P_3 = 0.05,\ P_2 = P_4 = 0.0,\ P_L = 0.001$

$\gamma_1 = \gamma_3 = 2.6,\ \alpha_1 = \alpha_3 = 0.28$

$\gamma_{xx} = 0.6,\ \gamma_{xy} = 0.4,\ \gamma_{xz} = 0.0$

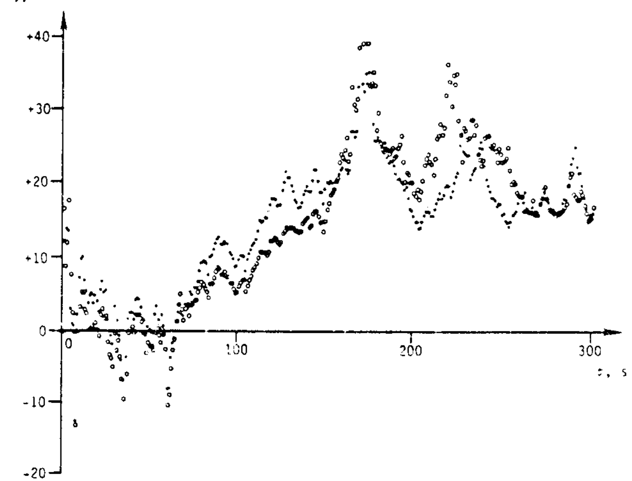
$\beta = 0.08115$

Magnetic reference antenna

$f_{samp} = 100$ Hz, $B = 1$ Hz

Seeds: 16379, 16656

Figure 4-6. Effect of quantization upon ANC improvement, X axis, high-level noise.



· ANC ∘ FNC

$P_1 = P_3 = 0.05,\ P_2 = P_4 = 0.0,\ P_L = 0.001$

$\gamma_1 = \gamma_3 = 2.6,\ \alpha_1 = \alpha_3 = 0.28$

$\gamma_{xx} = 0.6,\ \gamma_{xy} = 0.4,\ \gamma_{xz} = 0.0$

$\beta = 0.08115$

Magnetic reference antenna

$f_{samp} = 100$ Hz, $B = 1$ Hz

Seeds: 16379, 16656

Figure 4-7. Effect of cancellation methods, moderate-level noise, X a a decibel, especially for Gaussian noise and low-level atmospheric noise.

For moderate- and high-level atmospheric noise, somewhat larger differences occur, especially in the ANC-processing gain. The principal cause of the increased differences appears to be clipping of large-amplitude noise spikes by the A/D converter. It is apparent in Table 4-1 that the estimation error for linear processing is nearly identical for Gaussian and low-level atmospheric noise, but differs for moderate- and high-level atmospheric noise.

Analog noise was digitized with an A/D-converter gain of 2.0, which results in a converter clipping level of 2.5 V (Chapter 2). Since the maximum voltage recordable on the TEAC R-61 is 1.5 V, the A/D converter introduced no additional clipping of the ELF noise.

4.3 DIGITIZED-NOISE PROCESSOR

The NLP/ANC-processor program for digitized ELF noise was obtained by splitting the simulation program into two programs (Appendix C).

The first of these two programs generates simulated ELF noise as in the single-channel program. The noise is then quantized, packed, and written to disk in the format used to store the digitized ELF noise.

The second program reads digitized, packed noise data from a disk file. The data are then unpacked, converted to floating-point numbers, and processed as in the simulation program.

The program for processing digitized noise includes a subroutine for processing calibration signals when they are present. The operator must first use the file-display program (Chapter 2) to determine the duration (in seconds) and approximate voltage level of the 1-V calibration signal. The subroutine reads and decodes the data file for the specified number of seconds. Voltages falling within ten (10) percent of the specified value during the specified time interval are averaged to estimate the true level of the 1-V calibration signal in each channel. All subsequent voltages are divided by the appropriate averaged calibration voltage to scale them to true voltages.

Proper operation of these programs was verified by comparing their results with those obtained by the single-channel simulation program.

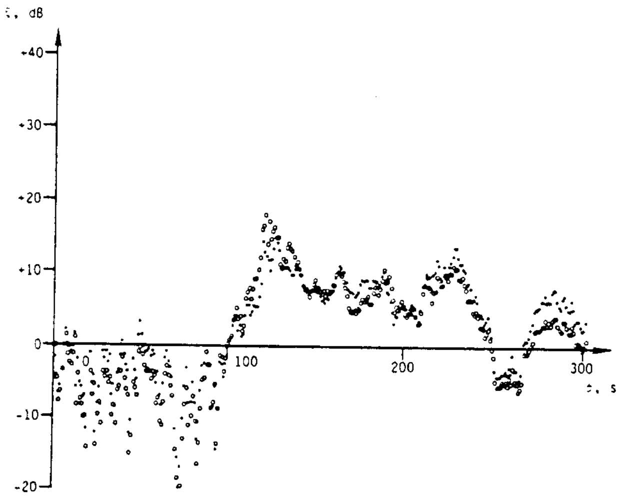
4.4 FIXED CANCELLATION

A fixed-noise-cancellation (FNC) processor was added to enable comparison of adaptive- and nonadaptive-cancellation techniques. The linear FNC processing occurs after independent nonlinear processing is applied to both the primary and reference inputs. The output from a selected axis of the reference antenna is then subtracted from the output of the primary antenna.

The results from one set of simulations with moderate-level atmospheric noise are shown in Figures 4-7 - 4-9. Convergence patterns are quite similar for both ANC and FNC. ANC usually (but not always) produces slightly more noise reduction than FNC. However, this simulation does not show the true advantages of ANC over FNC, since it employs identically oriented antennas and identical ground filters at each site.

4.5 REFERENCE

1. F. H. Raab, "Adaptive-noise-cancellation techniques for through-the-earth electromagnetics, Volume II," Final Report GMRR TR83-1, Green Mountain Radio Research Company, Winooski, VT, September 1983.



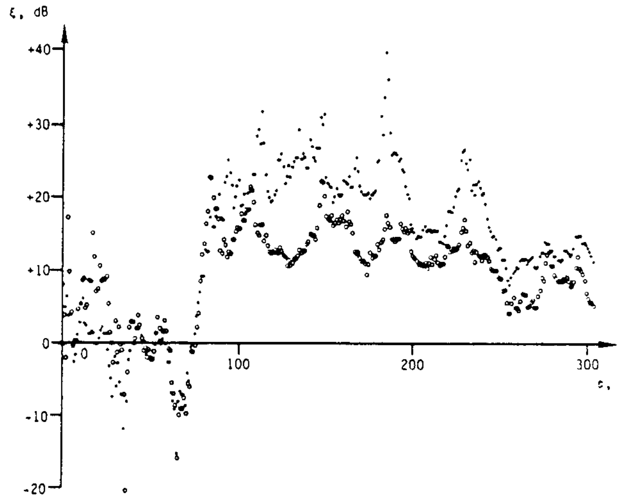
· ANC ∘ FNC

$P_1 = P_3 = 0.05,\ P_2 = P_4 = 0.0,\ P_L = 0.001$

$\Upsilon_1 = \Upsilon_3 = 2.6,\ a_1 = a_3 = 0.28$

$Y_{zx} = 0.6,\ Y_{zy} = 0.4,\ Y_{zz} = 0.0$
$s = 0.08115$
Magnetic reference antenna
$f_{samp} = 100$ Hz, $B = 1$ Hz
Seeds: 16379, 16656

Figure 4-8. Effect of cancellation methods, moderate-level noise, *Y* axis.



· ANC ∘ FNC

$P_1 = P_3 = 0.05,\ P_2 = P_4 = 0.0,\ P_L = 0.001$

$\Upsilon_1 = \Upsilon_3 = 2.6,\ a_1 = a_3 = 0.28$

$Y_{zx} = 0.6,\ Y_{zy} = 0.4,\ Y_{zz} = 0.0$
$s = 0.08115$
Magnetic reference antenna
$f_{samp} = 100$ Hz, $B = 1$ Hz
Seeds: 16379, 16656

Figure 4-9. Effect of cancellation methods, moderate-level noise, *Z* axis.

CHAPTER 5. RESULTS

Data from eleven of the fourteen ELF-noise tapes were used to evaluate the performance of the nonlinear-processing (NLP) and adaptive-noise-cancellation (ANC) algorithms. The characteristics of the data are discussed in Chapter 3, and the processing algorithms are described in Chapter 4. This chapter presents the results of the evaluation.

5.1 USEABLE DATA

Three of Develco's tapes (#1, #3, and #13) were "disturbed" by anomolies such as communication outages and cows rubbing against one of the receiving antennas. Since specific times of the disturbances are not known, data from these three tapes were not used in assessing average performance of the signal-processing algorithms.

The data-transfer procedure divided each analog tape into three useable disk files, each containing 540 s of ELF-noise data. Since obtaining reliable average-performance figures requires a large number of processing runs, each of the three useable disk files is subdivided into two 270-s runs. This division produces a total of 66 processing runs (24 X-axis, 18 Y-axis, and 24 Z-axis).

Because the fourth disk file for each tape contains no more than 120 s of data and the end of the tape creates a "disturbance," the fourth disk files are not processed. Since data set #9-1 contains a very long calibration signal, it is divided into two 260-s runs.

5.2 CONVERGENCE

The convergence of the NLP/ANC algorithm with real ELF noise is similar to that with simulated ELF noise. Examples of "normal" convergence trajectories are shown in Figures 5-1 - 5-3.

Convergence trajectories are, however, highly varied. For example, the trajectories of Figures 5-1 and 5-4 are derived from adjacent data segments of data set #8-1. An example of a "disturbance" and the algorithm's recovery can be seen in Figure 5-5.

5.3 PROCESSING GAIN

The averages for each processing run are assembled into a file, which is then processed to obtain averages, standard deviations, minima, and maxima. These results are computed as applicable for each noise tape, each axis, and overall. The results for individual data sets are presented in Appendix D. A number of tests (not shown) were also conducted to verify that the signal amplitude has an insignificant effect upon the signal-processing gains.

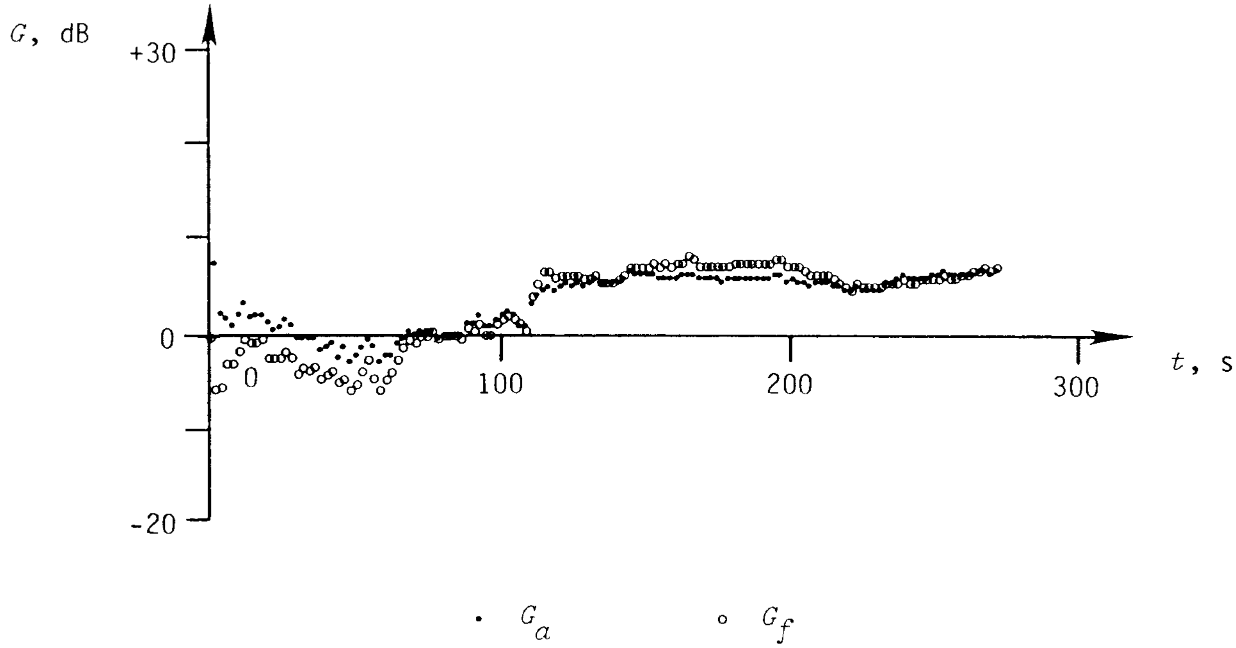


Figure 5-1. Convergence plot for data set #8-1A (X axis).

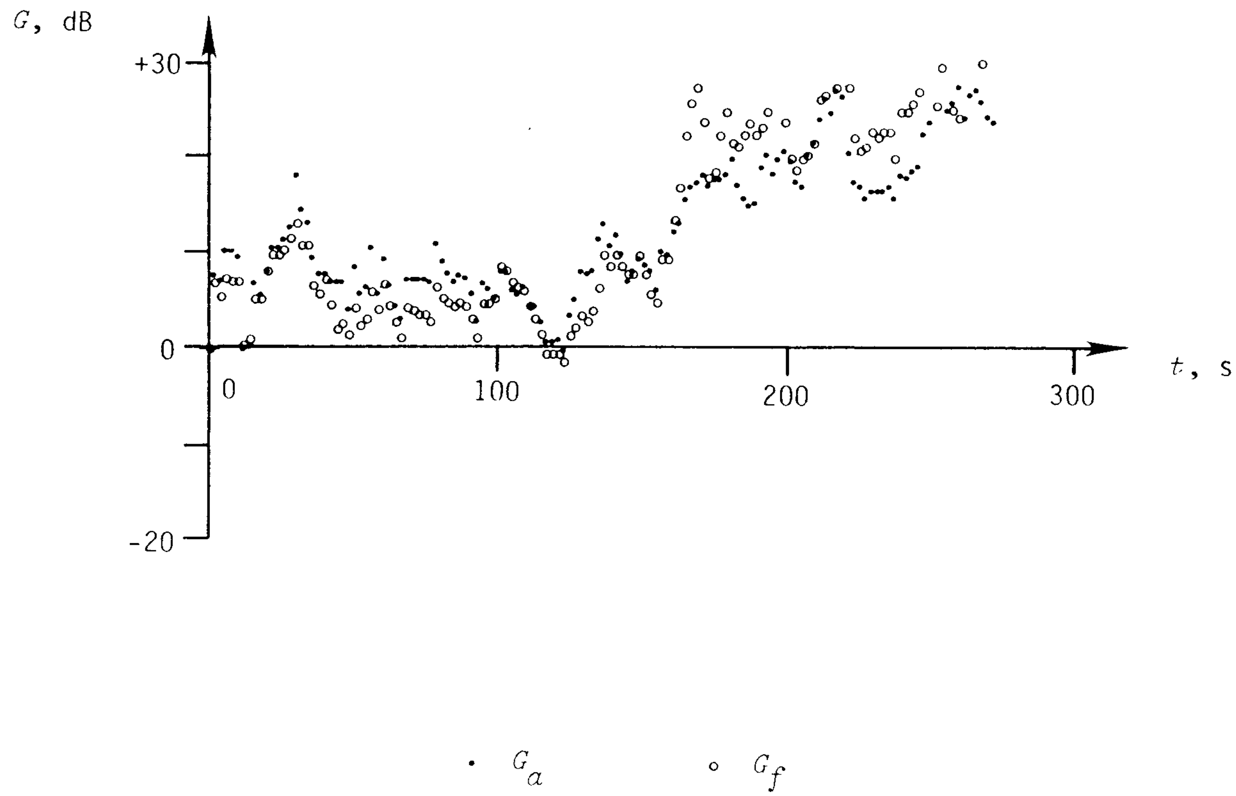


Figure 5-2. Convergence plot for data set #12-2A (Y axis).

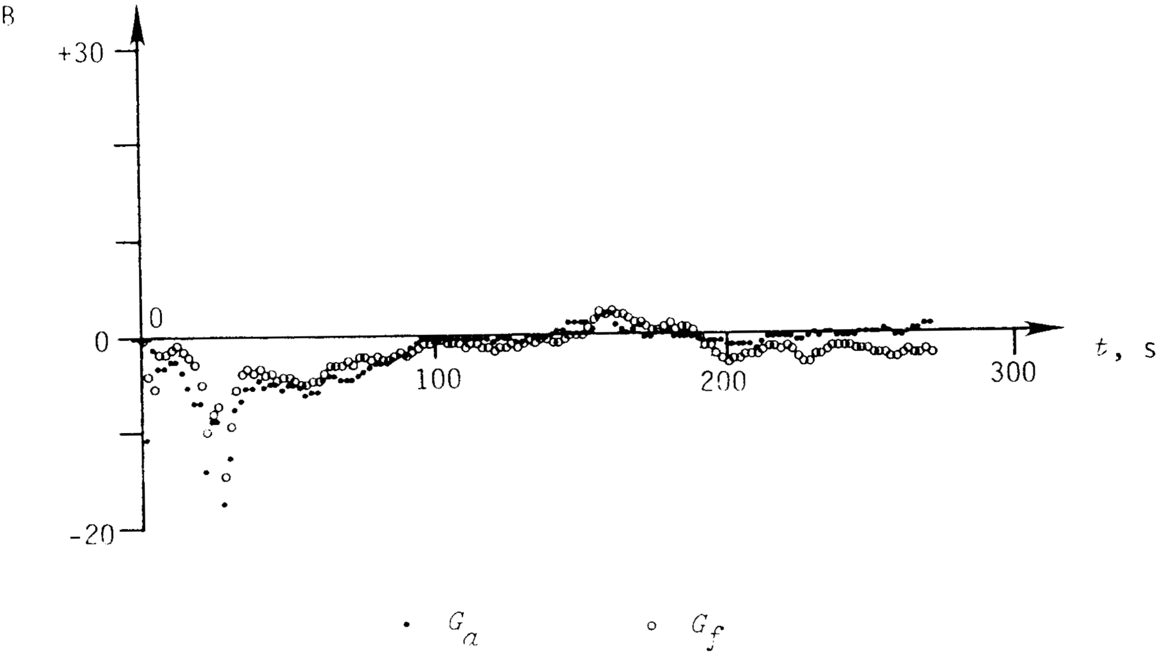


Figure 5-3. Convergence plot for data set #6-2A ($\dot{Z}$ axis).

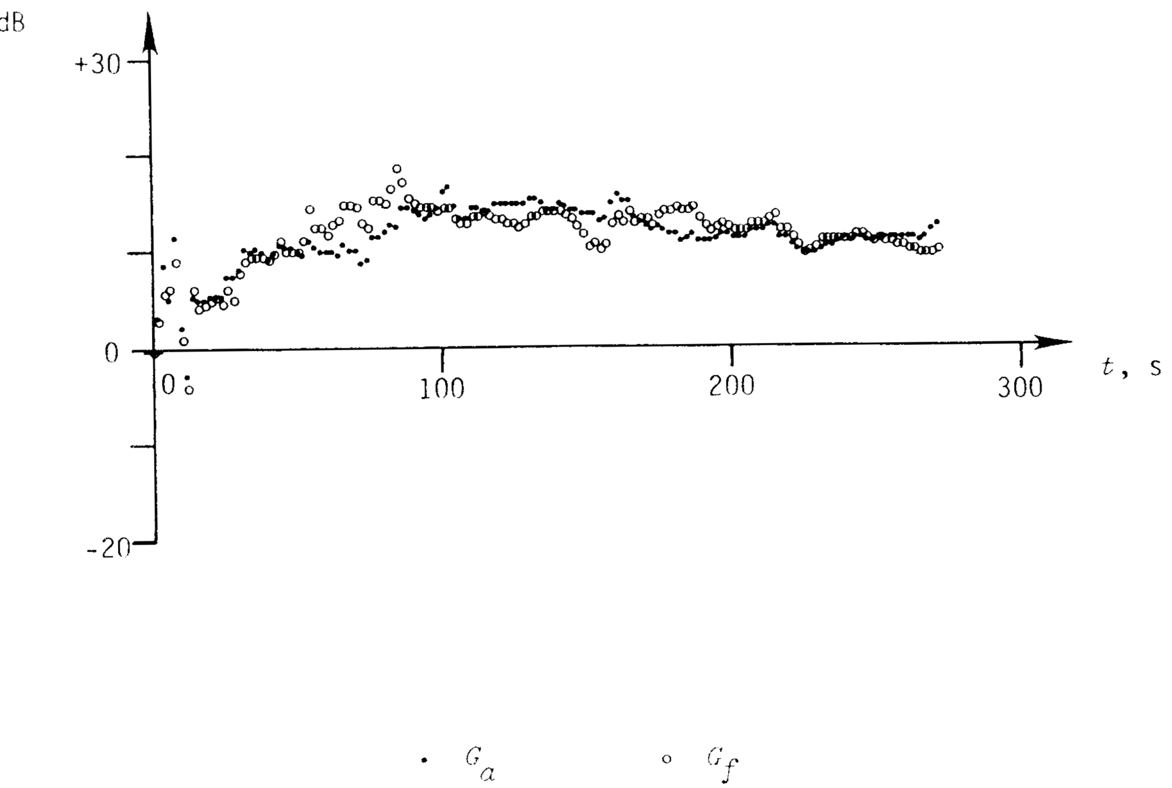


Figure 5-4. Convergence plot for data set #8-1B (X axis).

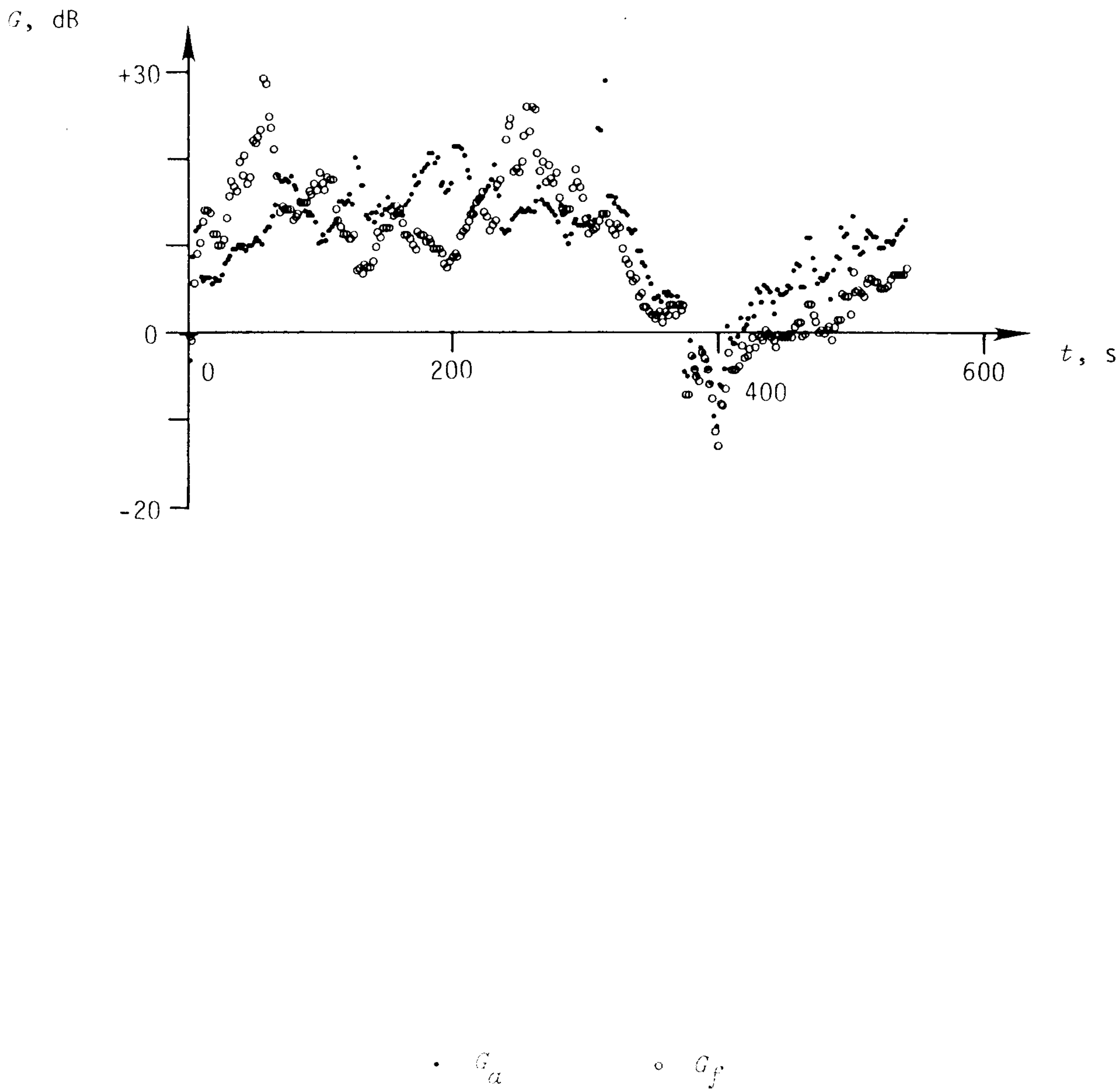


Figure 5-5. Convergence plot for "disturbed" data set #1-3 (Y axis).

The overall results are shown in Table 5-1 and Figures 5-6 - 5-11. Figure 5-6 shows the signal-processing gain achieved by nonlinear processing and ANC as a function of the adaptive-clipped threshold τ. NLP gain G_n is, by itself, insignificant. ANC gain G_a reaches a maximum value of 4.1 dB with $\tau = \infty$ to disable nonlinear processing. Setting τ to finite values (e.g., the 0.707 expected to optimize processing for moderate-level atmospheric noise) results in considerable degradation of G_a and total signal-processing gain G_{ta}.

Average signal-processing gains for each axis are shown in Figures 5-7 - 5-9 and compared in Figure 5-10. For the X axis, performance is similar to the all-axis performance, with G_a reaching a maximum value of 4.5 dB. For the Y axis, the maximum value of G_a is 11.9 dB; however, for the Z axis, the maximum value of G_a is -1.9 dB (a 1.9-dB *increase* in estimation error). Some nonlinear processing produces a very slight increase in total signal-processing gain for the X axis; however, in general nonlinear processing only reduces the total gain.

Histograms of the average ANC gains are presented in Figures 5-11 - 5-14. It is apparent from the histograms (and the standard deviations in Table 5-1) that there is a good deal of variability in the average gains. This is especially true for the all-axis averages, which are based upon inherently different individual-axis averages. For example, the standard deviation of the 4.1-dB all-axis average gain is 7.1 dB, whereas the standard deviation of the 11.9-dB Y-axis average gain is 6.1 dB. The averages are therefore significant but their values must not be considered exact.

The gains achieved by adaptive noise cancellation (ANC) and fixed noise cancellation (FNC) are compared in Figure 5-15. Overall, ANC is superior to FNC by 0.3 dB. However, for the Y axis (which has more noise power to be cancelled), G_a exceeds G_f by 1.6 dB. A gain of 1.6 dB is equivalent to a 30-percent reduction in signal-integration time. Under adverse conditions, G_a values as low as -6.7 dB are observed; in contrast, G_f values as low as -15.0 dB are observed. Under good conditions, G_a can be as high as 26.2 dB, while the maximum observed value of G_f is 22.6 dB.

5.4 WEIGHTING VECTOR

The average of the ANC weighting vectors is depicted in Figure 5-16 and Table 5-2. It is apparent that cancellation of the noise in the X and Y axes relies heavily upon the X and Y axes of the remote-reference antenna. However, the use of the other axes is not insignificant. The weighting vector for cancellation of Z-axis noise varies greatly and on the average is distributed evenly among all three remote-reference inputs.

Statistical analysis of disk-ANC numerical data, Program # FHR-354-

NUMERICAL DATA - AVERAGES

Single-channel NLP/ANC algorithm with disk-file input
Program # FHR-353E

Noise-file parameters: A/D gain = 2.0 Sampling frequency = 100.0
Total seconds in noise file = 575

Processing parameters: Signal = .1414 +j 0.000
Nsec = 270 Kout = 2 Nrun = 2
Nset = 2 F3dB = 1.00

Threshold = infinite

Set Axis Date Time Comment NSC NSS	Gn	Ga	Gta	Gf	Gtf
AVERAGES					
X axis (24 data sets)	0.0	4.3	4.3	3.6	3.6
Y axis (18 data sets)	0.0	11.9	11.9	10.3	10.3
Z axis (24 data sets)	0.0	-1.9	-1.9	-.8	-.8
Overall (66 data sets)	0.0	4.1	4.1	3.8	3.8
STANDARD DEVIATIONS					
X axis (24 data sets)	0.0	5.2	5.2	7.8	7.8
Y axis (18 data sets)	0.0	6.1	6.1	5.5	5.5
Z axis (24 data sets)	0.0	2.4	2.4	2.4	2.4
Overall (66 data sets)	0.0	7.1	7.1	7.1	7.1
MINIMA					
X axis (24 data sets)	0.0	-6.5	-6.5	-15.0	-15.0
Y axis (18 data sets)	0.0	4.2	4.2	1.4	1.4
Z axis (24 data sets)	0.0	-6.7	-6.7	-6.4	-6.4
Overall (66 data sets)	0.0	-6.7	-6.7	-15.0	-15.0
MAXIMA					
X axis (24 data sets)	0.0	12.6	12.6	13.9	13.9
Y axis (18 data sets)	0.0	26.2	26.2	22.6	22.6
Z axis (24 data sets)	0.0	2.9	2.9	5.3	5.3
Overall (66 data sets)	0.0	26.2	26.2	22.6	22.6

Threshold = 2.828

	Gn	Ga	Gta	Gf	Gtf
AVERAGES					
X axis (24 data sets)	.1	4.2	4.2	3.6	3.7
Y axis (18 data sets)	.3	10.7	11.0	5.1	5.4
Z axis (24 data sets)	-1.2	-1.5	-2.7	-1.0	-2.2
Overall (66 data sets)	-.3	3.9	3.6	2.3	2.0
STANDARD DEVIATIONS					
X axis (24 data sets)	.6	5.6	5.8	7.9	8.1
Y axis (18 data sets)	1.6	7.2	6.6	6.9	6.3
Z axis (24 data sets)	2.7	2.3	2.7	2.0	2.8
Overall (66 data sets)	1.9	7.1	7.5	6.5	6.9
MINIMA					
X axis (24 data sets)	-1.6	-6.6	-8.2	-14.3	-15.9
Y axis (18 data sets)	-1.2	-5.2	1.1	-7.7	-7.5
Z axis (24 data sets)	-6.4	-5.1	-7.0	-4.4	-6.1
Overall (66 data sets)	-6.4	-6.6	-8.2	-14.3	-15.9
MAXIMA					
X axis (24 data sets)	1.6	12.9	13.1	14.7	15.1
Y axis (18 data sets)	6.3	25.3	25.2	12.8	12.7
Z axis (24 data sets)	4.4	3.3	3.2	3.2	3.2
Overall (66 data sets)	6.3	25.3	25.2	14.7	15.1

Note: Data sets #1, #3, and #13 excluded due to distur

Table 5-1. Processing-gain statistics.

Threshold = 1.414

Set Axis Date Time	Comment	NSC NSS	Gn	Ga	Gta	Gf	Gtf
X axis (24 data sets)			.4	4.2	4.5	2.8	3.1
Y axis (18 data sets)			.9	9.0	9.9	7.4	8.3
Z axis (24 data sets)			-1.1	-1.6	-2.6	-1.3	-2.4
Overall (66 data sets)			-.0	3.4	3.4	2.5	2.5
STANDARD DEVIATIONS							
X axis (24 data sets)			2.9	6.2	6.0	9.0	9.0
Y axis (18 data sets)			2.5	7.3	6.6	6.8	5.9
Z axis (24 data sets)			4.3	2.6	4.0	3.0	4.1
Overall (66 data sets)			3.5	6.9	7.4	7.5	7.9
MINIMA							
X axis (24 data sets)			-6.4	-8.2	-3.6	-11.6	-17.6
Y axis (18 data sets)			-1.3	-8.8	-7.3	-8.9	-7.1
Z axis (24 data sets)			-9.2	-5.3	-9.7	-7.0	-8.9
Overall (66 data sets)			-9.2	-8.8	-9.7	-11.6	-17.6
MAXIMA							
X axis (24 data sets)			7.8	19.5	20.9	21.5	22.9
Y axis (18 data sets)			9.0	19.1	19.8	17.2	19.4
Z axis (24 data sets)			8.1	3.8	7.0	3.8	4.9
Overall (66 data sets)			9.0	19.5	20.9	21.5	22.9

Threshold = 0.707

Set Axis Date Time	Comment	NSC NSS	Gn	Ga	Gta	Gf	Gtf
X axis (24 data sets)			-.4	2.9	2.5	1.0	.6
Y axis (18 data sets)			2.1	5.8	7.8	3.4	5.5
Z axis (24 data sets)			-.9	-1.6	-2.6	-2.5	-3.5
Overall (66 data sets)			.1	2.0	2.1	.4	.5
STANDARD DEVIATIONS							
X axis (24 data sets)			4.0	5.6	6.9	7.1	8.9
Y axis (18 data sets)			3.7	5.5	4.3	6.6	4.6
Z axis (24 data sets)			5.1	2.4	4.3	4.4	4.7
Overall (66 data sets)			4.5	5.5	6.8	6.5	7.3
MINIMA							
X axis (24 data sets)			-10.3	-8.5	-6.9	-14.5	-21.2
Y axis (18 data sets)			-2.8	-8.0	-1.0	-10.6	-3.7
Z axis (24 data sets)			-10.3	-6.2	-10.7	-12.9	-10.3
Overall (66 data sets)			-10.3	-8.5	-10.7	-14.5	-21.2
MAXIMA							
X axis (24 data sets)			6.1	15.8	19.5	11.5	14.7
Y axis (18 data sets)			8.8	13.0	14.7	14.7	15.9
Z axis (24 data sets)			9.3	3.1	5.9	6.4	5.4
Overall (66 data sets)			9.3	15.8	19.5	14.7	15.9

Note: Data sets #1, #3, and #13 excluded due to disturbances.

Table 5-1. Processing-gain statistics. (Continued)

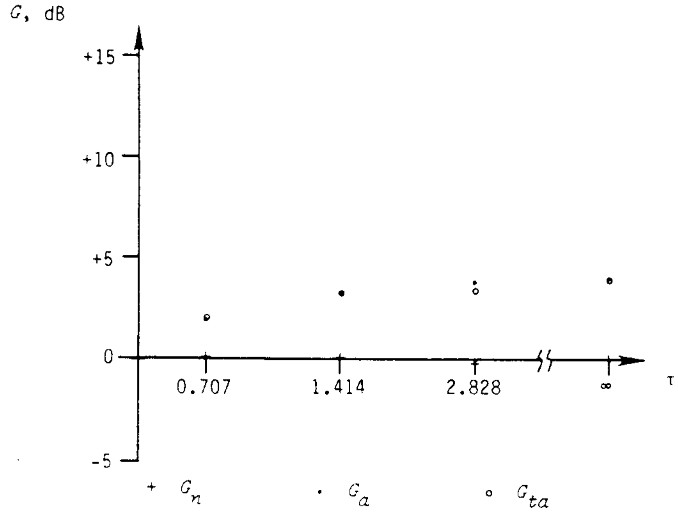


Figure 5-6. Average ANC gains, all axes.

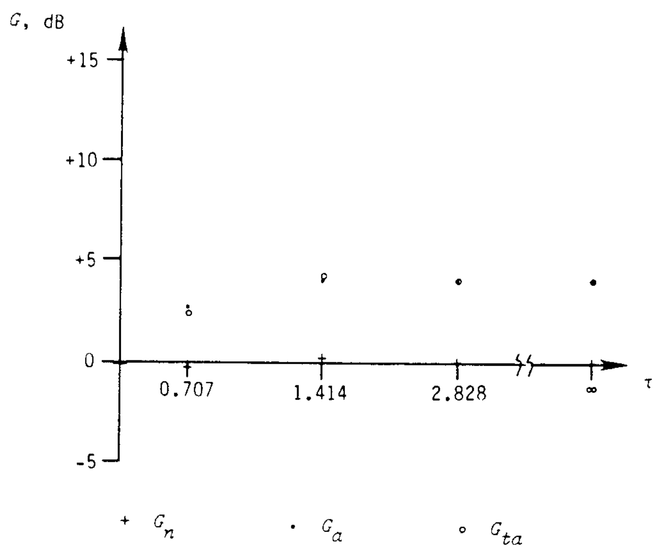


Figure 5-7. Average ANC gains, *X* axis.

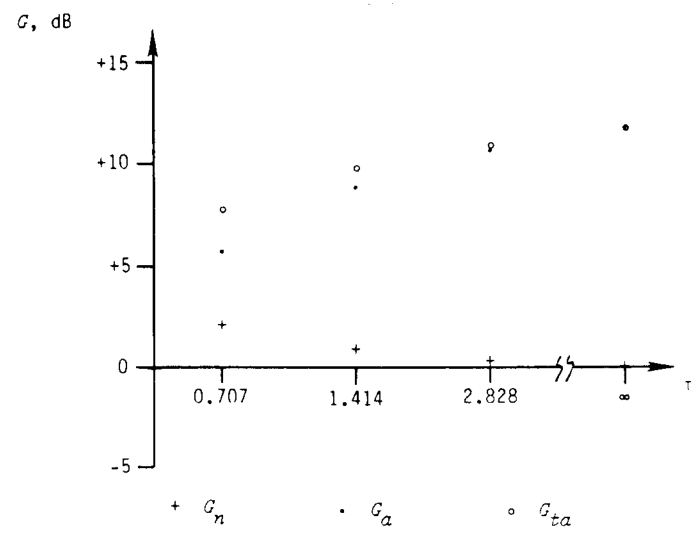


Figure 5-8. Average ANC gains, *Y* axis.

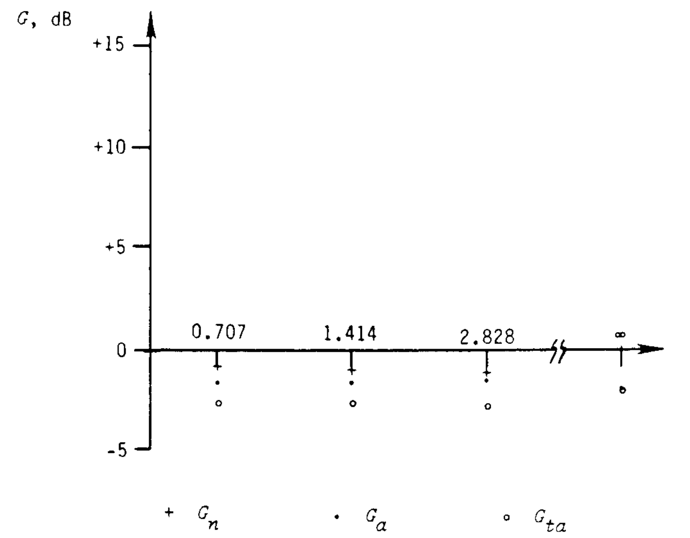


Figure 5-9. Average ANC gains, *Z* axis.

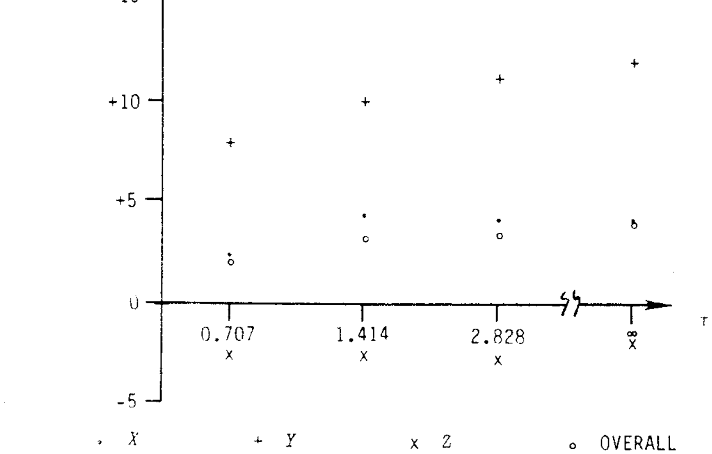
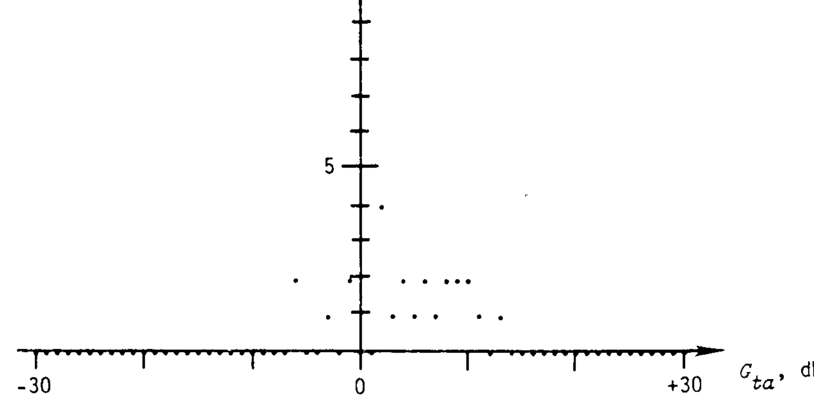
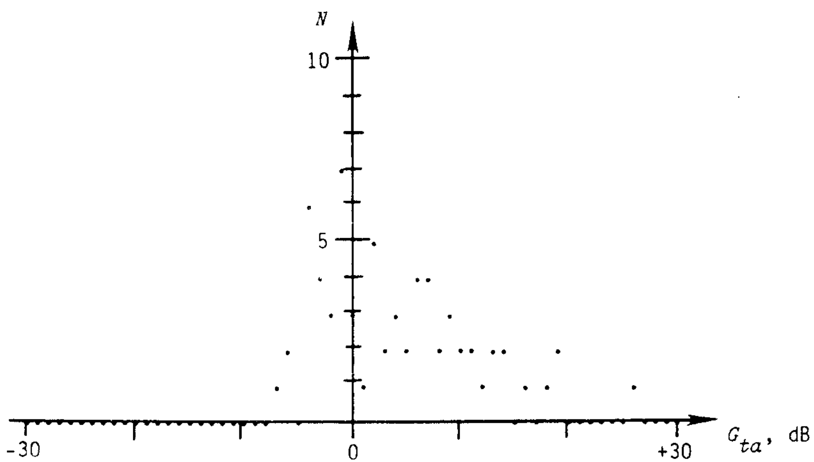


Figure 5-10. Comparison of average total ANC/NLP gain by axis.



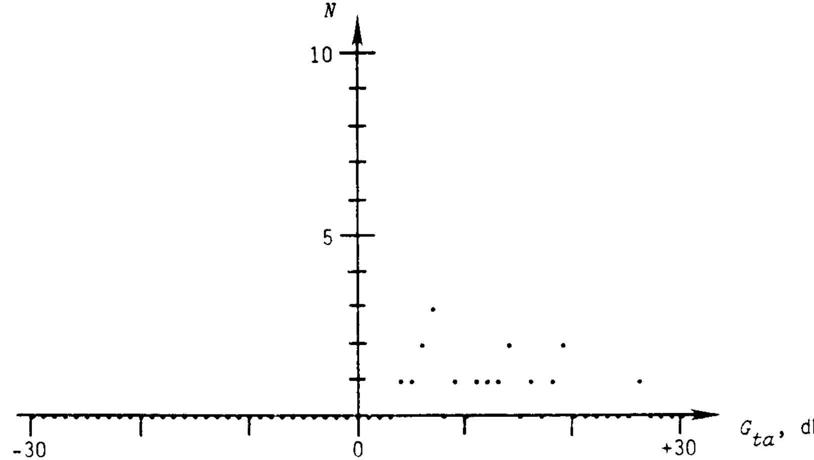
Interval width: 1.0 Total number of samples = 24 $\tau = \infty$

Figure 5-12. Histogram of average processing gain, X axis.



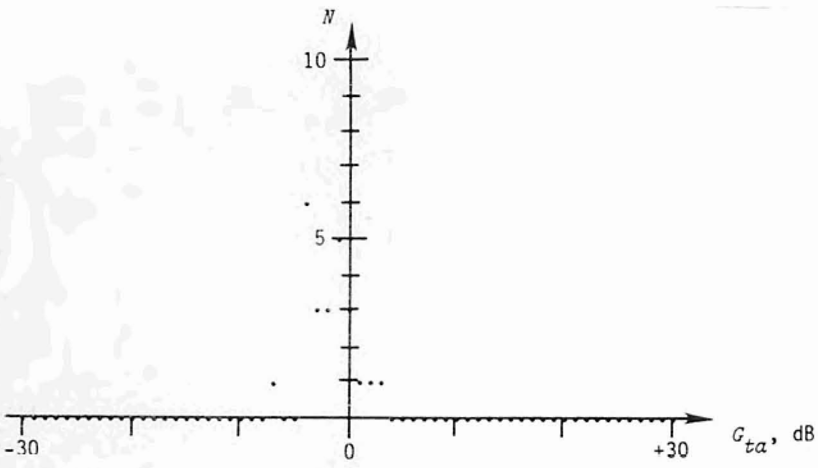
Interval width: 1.0 Total number of samples = 66 $\tau = \infty$

Figure 5-11. Historgram of average processing gain, all axes.



Interval width: 1.0 Total number of samples = 18 $\tau = \infty$

Figure 5-13. Histogram of average processing gain, Y axis.



Interval width: 1.0 Total number of samples = 24 $\tau = \infty$

Figure 5-14. Histogram of average processing gain, Z axis.

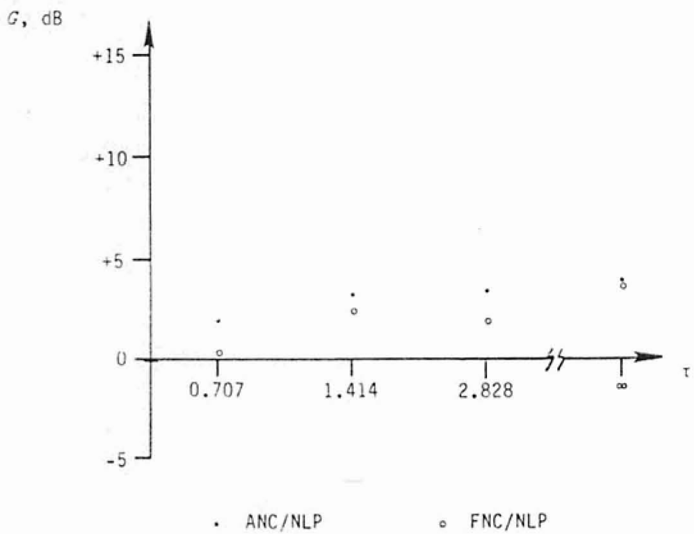


Figure 5-15. Comparison of average total ANC/NLP and FNC/NLP gains.

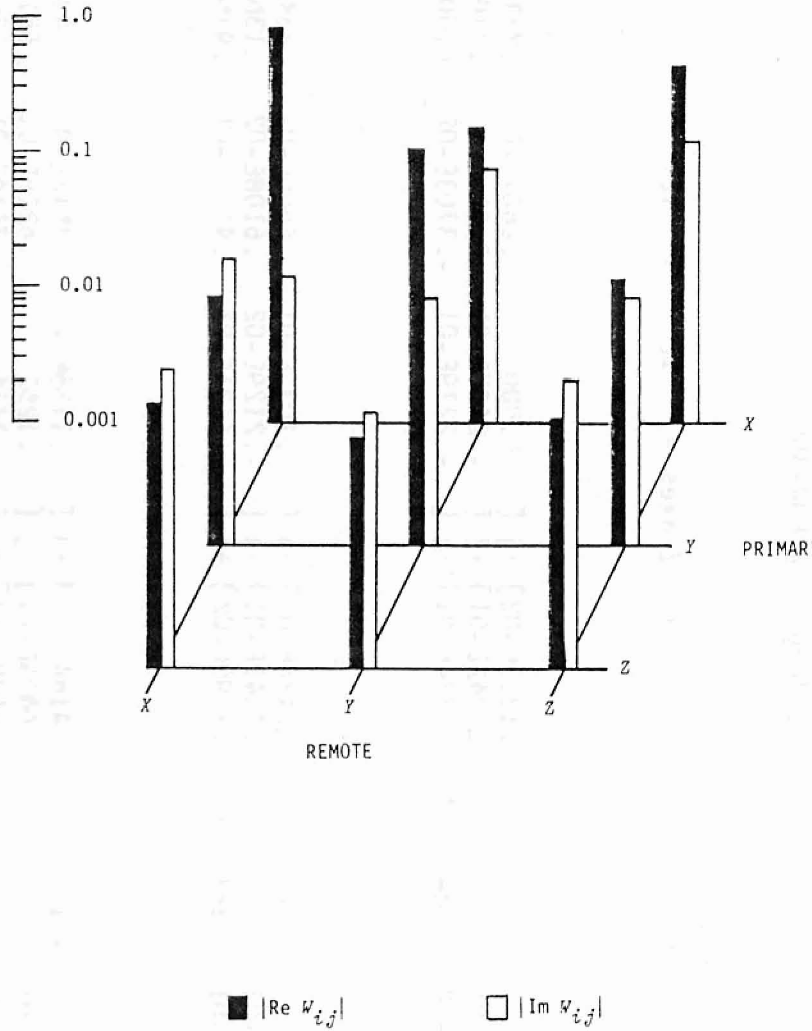


Figure 5-16. Average weighting **W**.

Averaging of correlation data, Program # FHR-355-

Total number of data sets = 33
Numbers of samples for local X, Y, and Z axes = 12 9 12

Ryy AVG =

```
[   .2452       -.6552E-01   -.1367E-02] +j [   0.000        .2607E-01    .2219E-01]
[  -.6552E-01    .3279        .3842E-01] +j [  -.2607E-01   0.000         .3703E-02]
[  -.1367E-02    .3842E-01    .2774E-01] +j [  -.2219E-01  -.3703E-02    0.000    ]
```

Rxy AVG =

```
[   .2003       -.1074       -.2424E-01] +j [   .1033E-01    .4606E-01    .2348E-01]
[  -.7892E-01    .2720        .2841E-01] +j [  -.2129E-02    .6108E-02    .1380E-03]
[   .1760E-01    .1649E-01    .4590E-02] +j [  -.4121E-01    .4138E-01    .9358E-02]
```

W AVG =

```
[   .8024       -.1324       -.4164    ] +j [   .1185E-01    .7512E-01    .1435    ]
[  -.6374E-01    .8018       -.8835E-01] +j [   .1252        .6226E-01   -.6517E-01]
[   .8686E-01    .4911E-01    .6538E-01] +j [  -.1475        .7414E-01    .1293    ]
```

Table 5-2. Average covariance, cross correlation, and weighting.

5.5 NOISE CORRELATION

The output file produced from the noise-data processing program (Appendix C) includes the estimated remote-reference covariance matrix $\hat{R}_{yy}$, cross-correlation vector $\hat{\boldsymbol{r}}_{xy}$, and weighting vector $\boldsymbol{w}$ for each run. These data were assembled into a file and then averaged.

The resultant average covariance and cross-correlation matrices are given in Table 5-2 and represented graphically in Figures 5-17 and 5-18. On the average, the Y-axis noise power exceeds the X-axis noise power by 1.3 dB and the Z-axis noise power by 10.7 dB. While the X-axis noise power exceeds the Y-axis noise power in some data sets, the Z-axis noise power is consistently about 10-dB lower than the X-axis and Y-axis noise powers.

Axis-to-axis correlation is readily observable in the Figures 5-17 and 5-18. To compare correlations, it is necessary to normalize the off-diagonal terms of $\hat{R}_{yy}$ by the corresponding noise powers; e.g.,

$$\hat{\rho}_{xy} = \frac{R_{yy}(x,y)}{[R_{yy}(x,x)\ R_{yy}(y,y)]^{1/2}} \ . \tag{1}$$

The results thus obtained from the values in Table 5-2 are:

$$\begin{aligned} \hat{\rho}_{xy} &\cong 0.25 \\ \hat{\rho}_{xz} &\cong 0.27 \ . \\ \hat{\rho}_{yz} &\cong 0.40 \end{aligned} \tag{2}$$

The cross-correlation matrix (Figure 5-18) is generally similar to the covariance matrix (Figure 5-17). The lack of Z-axis-to-Z-axis correlation is apparent. Phase shift in the X-X and Y-Y elements is also apparent. The magnitude of the quadrature component of these elements is a measure of the effect of the local geology. It is interesting to note that the magnitudes of the imaginary part of the X-X and Y-Y elements is comparable to the magnitude of the Z-Z element. This is not suprising, since the Z-axis magnetic field is produced by the local geology.

5.6 ANALYSIS

The ANC algorithm works as it should, reducing the effective noise level by 4.1 dB on the average. The improvement is greater when the noise level is higher (e.g., 11.9 dB for the Y axis) and lower when the noise level is lower (e.g., -1.9 dB for the Z axis).

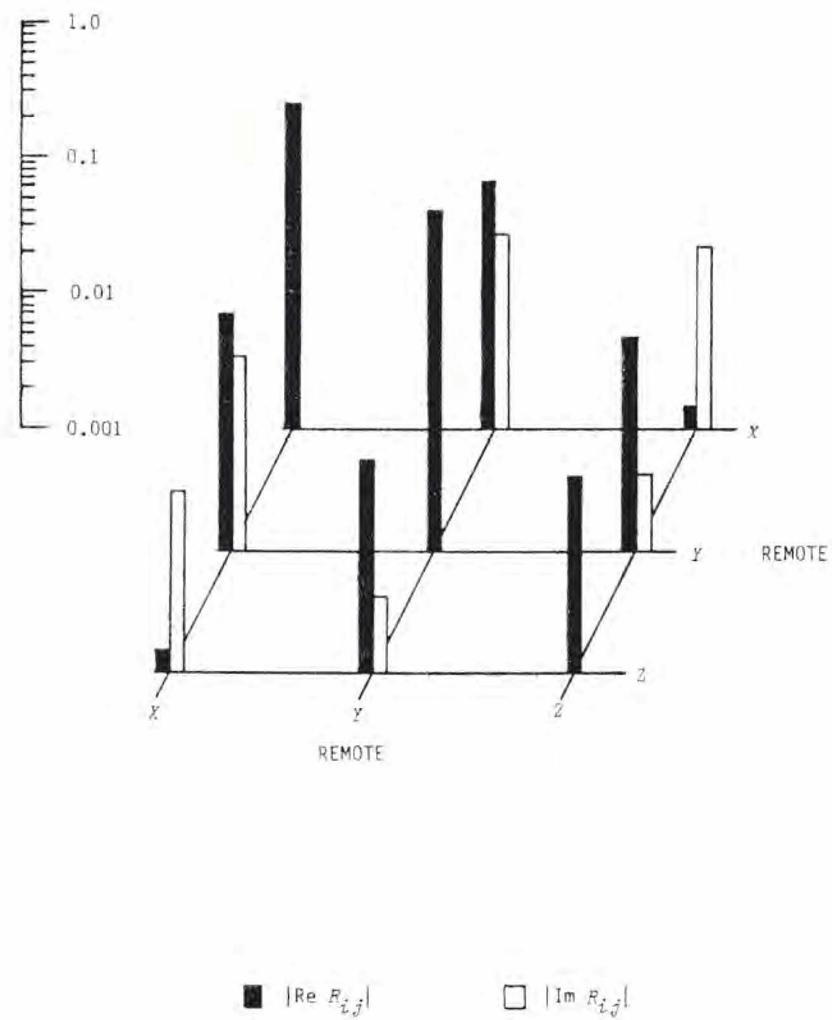


Figure 5-17. Average noise covariance R_{yy}.

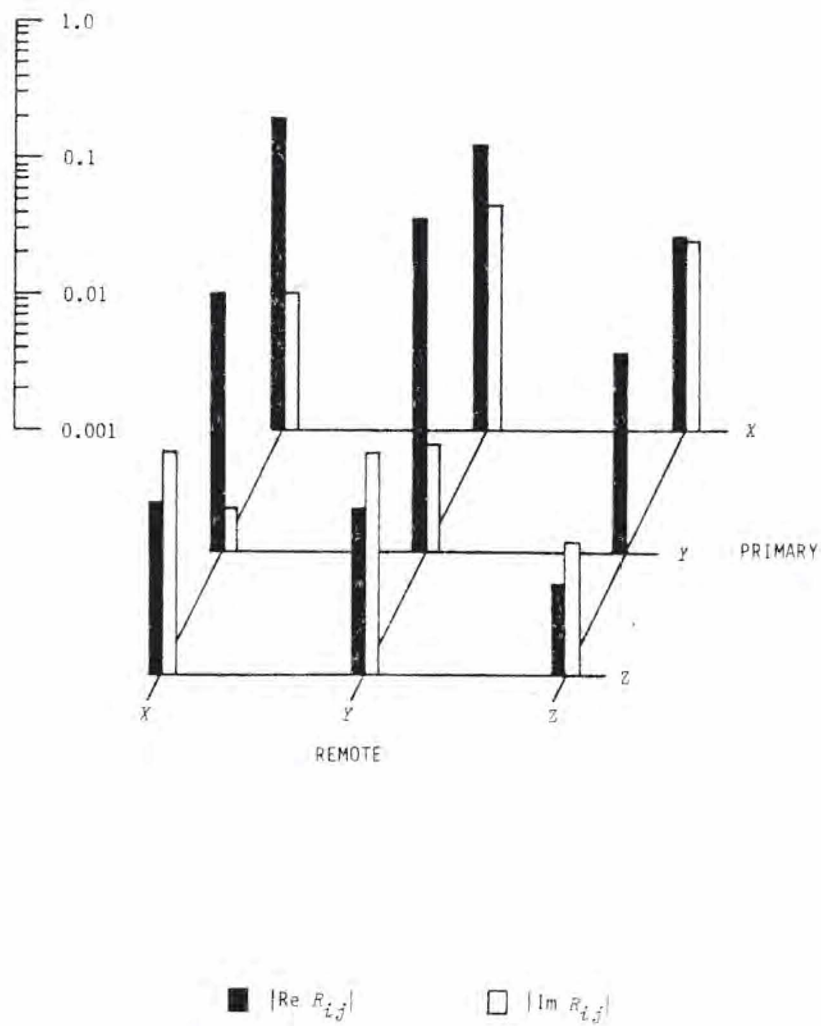


Figure 5-18. Average noise cross-correlation R_{xy}.

Overall, the ANC gain exceeded the FNC improvement by only 0.3 dB. However, as atmospheric noise power (hence correlation) increases, the difference between ANC and FNC gains also increases. The most significant advantage of ANC is its ability to prevent disastrous SNR degradation under adverse conditions. ANC is therefore definitely desirable.

Nonlinear processing was on the average harmful for this set of noise data. The gain G_n was relatively small and more than overcome by the associated decrease in G_a.

The character of the noise should be determined from statistical analysis. However, some conclusions about its character can be drawn from the behavior of the ANC and NLP algorithms. The small or negative gains achieved by nonlinear processing suggest that the Z-axis noise is Gaussian and that the X- and Y-axis noises are only slightly impulsive (not even "low-level," however). The observed correlation matrices and ANC gains are consistant with local, Gaussian noise whose amplitude is 10-dB below that of the atmospheric noise. These observations on the character of the noise are consistant with those made by Develco engineers (Chapter 3).

Because of the relatively high values of the standard deviations, the averages derived here must be considered approximate indications of what is to be expected. A considerably greater number of noise-data sets will be required to improve the accuracy of the averages.

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS

The extension of through-the-earth electromagnetic location techniques to deep-mine depths (1-km) requires more noise reduction than can be achieved (in an acceptable amount of time) by simple signal integration. Intrinsic-safety requirements preclude significant increases in the power of the underground transmitter. The development of a signal-processing technique such as adaptive noise cancellation is therefore clearly necessary.

6.1 ACCOMPLISHMENTS

Phase I of this program was a preliminary theoretical investigation of ELF-noise characteristics and ANC techniques. The nature of the multicomponent ELF-noise model was defined. The available noise-correlation data were reviewed and the results suggested that from 10 to 40 dB of noise reduction could be achieved by the use of ANC. DMI and LMS techniques were identified as candidate ANC algorithms.

Phase II of this program was a comparison and evaluation of techniques through simulation. Simulation software was implemented for both the multicomponent ELF-noise model and the NLP/ANC signal processor. Nonlinear processing techniques were compared and the adaptive clipper was found to offer the greatest processing gain. Adaptive-noise-cancellation techniques were compared and DMI was found to be superior to LMS. Simulation results showed that it is desirable to include both NLP and ANC in the signal processor, but the gains of the two processors are not additive. Noise reductions ranging from 5 to 25 dB were predicted for typical conditions.

The objective of Phase III is to evaluate the signal-processing algorithm with real ELF noise. The principal accomplishments during Phase III of this program are:

- Implementation of hardware and software to digitize ELF noise,
- Digitization of fourteen half-hour analog tapes of ELF noise,
- Modification of the Phase-II signal-processing algorithm to work with a single-channel primary antenna and to accept disk-file input,
- Processing of the digitized data with various clipping thresholds, and
- Compilation and analysis of the results.

6.2 RESULTS AND CONCLUSIONS

The all-axis average processing gain is 4.1 dB. When processing gains are averaged on an axis-by-axis basis, ANC gains of 4.3, 11.9, and -1.9 dB are obtained for the X, Y, and Z axes, respectively. The ANC gain roughly follows the received noise power in each axis.

Nonlinear processing by itself produces a small gain in some cases. However, the resultant degradation of the ANC gain is far larger than the NLP gain. The greatest total gain was achieved with nonlinear processing disabled.

The number of data sets available is somewhat small, all data sets were recorded during the same season, and the noise levels were in general 15 to 20 dB lower than nominal. The results obtained should therefore be regarded as pessimistic approximate indications of what can be achieved. However, the results clearly show that ANC works. Under high-noise-power conditions, ANC can make a significant improvement in the SNR of through-the-earth electromagnetic systems operating at extremely low frequencies.

6.3 RECOMMENDATIONS

An effective signal-processing technique offering SNR enhancement in TTE EM systems has been developed by this program. However, there remain several unanswered questions and conjectures, as well as a need for more ELF noise data.

Determination of the characteristics of the digitized ELF noise was not part of the Phase-III efforts. From the performance of the signal-processing algorithm, the noise appears to be only slightly impulsive (somewhere between Gaussian and low-level impulsivity, in contrast to the anticipated moderate-level impulsivity). A statistical evaluation of the digitized noise would allow validation of the multicomponent model and determination of its parameters for use in future simulations. This evaluation should include determination of histograms, moments, best fit to the Field-Lewinstein-Modestino model, and direction-of-arrival characteristics.

The digitized noise is believed to represent quiet atmospheric conditions. Additional measurements made at various times throughout the year are required to ascertain what to anticipate for normal and worst-case operational environments.

Additional data sets would also enable a more accurate assessment of the capabilities of the signal-processing system. It is desirable (although not absolutely necessary) to have all three axes of the primary antenna recorded simultaneously.

The results presented in Chapter 5 show the potential adverse impact of an incorrect setting of the adaptive-clipper threshold. Disabling the nonlinear processing is not advisable, since operation may be required in the presence of thunderstorms or machine-generated impulsive noise. In addition, ELF noise is expected in general to be somewhat more impulsive than that used in this evaluation. A possible solution is to estimate both the second and fourth moments of the received noise and set the thresold to correspond to both noise power and impulsivity. Research pm this two-degree-of-freedom adaptive clipper is needed.

The signal-processing algorithm developed by this program employs a narrowband ANC algorithm. Nonlinear processing is applied independently to each wideband input, which is then averaged. The ANC processing then uses the estimated noise covariance and cross-correlation to combine the averages to minimize the expected estimation error.

An alternative approach employs a wideband ANC processor. In this approach, wideband inputs are combined by the ANC processor, whose output is then applied to the nonlinear processor. The wideband ANC system requires a recursive algorithm, since the matrix inversions required by the DMI algorithm make it unsuitable for real-time use. It is not anticipated that the wideband algorithm would make the ANC and NLP gains additive, since the ANC processor should reduce greatly the input to the nonlinear processor. Additional study and simulation are required to determine whether or not the performance of the wideband algorithm can be expected to be better than that of the narrowband algorithm developed by this program.

APPENDIX A. DIGITIZATION PROGRAMS

A/D-Interface Subroutines

```
C  Program # FHR-342C, 10/20/83
C  Interface subroutines for I/O technology A/D/A board
C
      SUBROUTINE ADVERS(LUN)
C  Print version number
      IMPLICIT LOGICAL (L)
      WRITE(LUN,10)
   10 FORMAT(' A/D/A interface, Program # FHR-342C')
      RETURN
      END


      SUBROUTINE ADSET(LCHAN,LGAIN,LSD,LBIAS,LCBU,LCBL)
C  Form control bytes and set addresses
C  Configured for access from memory, base address = ABASE
C  LSD =
C         0 for single-ended operation
C        32 for differential operation
C  LSR =  (applicable only if D/A is used)
C         0 for single
C        64 for refresh
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      LOGICAL PEEK
      DIMENSION LBIT(8)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
      DATA ABASE /X'EC00'/
      DATA LBIT / X'01', X'02', X'04', X'08',
     >            X'10', X'20', X'40', X'80' /
C
C  Addresses for upper and lower control bytes
C
      ACBU = ABASE + 1
      ACBL = ABASE
C
C  Form control bytes
C
      LCBU = LBIAS
      LCBL = LCHAN .OR. (LGAIN*8) .OR. LSD .OR. LSR
C
C  Add start-conversion bit
C
      LCBL = LCBL .OR. LBIT(8)
C
      RETURN
      END


      SUBROUTINE ADCNV(LCBU,LCBL,LR2,LR1)
C  Read A/D converter
C  Configured for access from memory
C  Must call ADSET prior to first use of ADCNV
C  Test result, North Star Horizon:  < 0.4 ms per reading
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      LOGICAL PEEK
      DIMENSION LBIT(8)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
C
C  Start first conversion
C
      CALL POKE(ACBU,LCBU)
      CALL POKE(ACBL,LCBL)
C
C  Read upper byte and test for settling
```

```
C
  100 LTEST = PEEK(ACBU) .AND. LBIT(8)
      IF(LTEST.EQ.0) GO TO 100
C
C  Start second conversion
C
      CALL POKE(ACBL,LCBL)
C
C  Read upper byte and test for completion
C
  200 LR2 = PEEK(ACBU)
      LTEST = LR2 .AND. LBIT(8)
      IF(LTEST.EQ.0) GO TO 200
C
C  Conversion completed, read lower byte
C
      LR1 = PEEK(ACBL)
C
      RETURN
      END


      SUBROUTINE ADVOLT(LR2,LR1,LGAIN,LCHAN,V)
C  Interpret output of A/D converter
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      REAL V
C
C  Get channel no
C
      LCHAN = (LR2 .AND. (X'70')) / 16
C
C  Get lower byte
C
      I1 = LR1 .AND. (X'7F')
      LTEST = LR1 .AND. (X'80')
      IF(LTEST.NE.0) I1 = I1 .OR. (X'0080')
C
C  Get upper byte
C
      LTEST = LR2 .AND. (X'07')
      I2 = LTEST
      I2 = I2 * 256
C
C  Assemble into integer
C
      IV = I1 .OR. I2
C
C  Get sign bit
C
      LSIGN = LR2 .AND. (X'08')
C
C  Convert to real
C
      V = 2047.5 - IV
      IF(LSIGN.EQ.0) V = V - 2048.0
C
C  Include full-scale and gain factors
C
      V = V * 4.8828327E-03 / 4**LGAIN
C
      RETURN
      END


      SUBROUTINE ADSIM(LGAIN,LCHAN,V,LR2,LR1)
C  Simulated A/D output bytes for specified gain, channel, and voltage
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      REAL V, VS
C
```

```
C  Check for saturation, then scale voltage
C
      VS = Ø.5 * V * 4**LGAIN
      IF(VS .GT.  4.9988) VS =  4.9988
      IF(VS .LT. -4.9988) VS = -4.9988
      VS = VS * 4Ø9.5983
C
C  Extract sign, convert to integer
C
      LSIGN = 1
      IF(VS.LT.Ø.Ø) LSIGN = Ø
      IF(VS.LT.Ø.Ø) VS = VS + 2Ø47.5
      IV = IFIX(2Ø47.5 - VS)
C
C  Form bytes
C
      IV = IV .AND. (X'Ø7FF')
      LR1 = IV .AND. (X'ØØFF')
      LR2 = (IV / 256) .OR. (LCHAN * 16)
      IF(LSIGN.EQ.1) LR2 = LR2 .OR. (X'Ø8')
C
      RETURN
      END


      SUBROUTINE ADBIAS(LCHAN,LGAIN,LSD,LBIAS2,VBIAS2)
C  Find bias constant required to zero A/D output
C  Ground or short inputs during zeroing
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      REAL VBIAS1, VBIAS2, VBIAS3
      LOGICAL PEEK
      DIMENSION LBIT(8)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
C
C  Set up end points and find corresponding readings
C
      NIT = Ø
      NITMAX = 1Ø
      IBIAS1 = Ø
      LBIAS1 = IBIAS1
      CALL ADSET(LCHAN,LGAIN,LSD,LBIAS1,LCBU,LCBL)
      CALL ADCNV(LCBU,LCBL,LR2,LR1)
      CALL ADVOLT(LR2,LR1,LGAIN,LCHD,VBIAS1)
      IBIAS3 = 255
      LBIAS3 = IBIAS3
      CALL ADSET(LCHAN,LGAIN,LSD,LBIAS3,LCBU,LCBL)
      CALL ADCNV(LCBU,LCBL,LR2,LR1)
      CALL ADVOLT(LR2,LR1,LGAIN,LCHD,VBIAS3)
C
C  Begin iteration
C
  1ØØ NIT = NIT + 1
      IBIAS2 = (IBIAS1 + IBIAS3) / 2
      LBIAS2 = IBIAS2
      CALL ADSET(LCHAN,LGAIN,LSD,LBIAS2,LCBU,LCBL)
      CALL ADCNV(LCBU,LCBL,LR2,LR1)
      CALL ADVOLT(LR2,LR1,LGAIN,LCHD,VBIAS2)
C
C  Select new endpoint or quit
C
      IF(NIT.EQ.NITMAX) GO TO 2ØØ
      IF(VBIAS2.GT.Ø) GO TO 13Ø
C
      IBIAS1 = IBIAS2
      LBIAS1 = LBIAS2
      VBIAS1 = VBIAS2
      GO TO 1ØØ
C
  13Ø IBIAS3 = IBIAS2
      LBIAS3 = LBIAS2
```

```
      VBIAS3 = VBIAS2
      GO TO 100
C
  200 RETURN
      END


      SUBROUTINE ADGAIN(LGAIN)
C  Set gain according to maximum intended input voltage
      IMPLICIT LOGICAL (L)
      DIMENSION G(4)
      DATA G /0.5, 2.0, 8.0, 32.0/
C
   10 WRITE(3,20)
   20 FORMAT(' Maximum input voltage = ')
      CALL ATR(VMAX,IERR)
      GMAX = 4.9988 / VMAX
      IF(GMAX.LT.G(1)) GO TO 10
C
      N = 2
   40 IF(GMAX.LT.G(N)) GO TO 80
      N = N + 1
      IF(N.EQ.5) GO TO 80
      GO TO 40
C
   80 N = N - 1
      LGAIN = N - 1
      WRITE(3,90) G(N)
   90 FORMAT(' Gain = ',F5.1)
C
      RETURN
      END
```

SemiDisk-Interface Subroutines

```
C  Program # FHR-330-, 07/28/83
C  Subroutines for direct access to SemiDisk
C  WARNING:  Cannot use these and SemiDisk as disk at same time!
C
C  Counters:
C     Byte:   0 - 127
C     Sector: 0 - 15
C     Track : 0 - 255
C


      SUBROUTINE SDINIT(LCLR)
C  Initialization for direct read-write to SemiDisk
C
C  Set LCLR = 0 to set parameters
C           = 1 to set parameters and clear SemiDisk
C
      IMPLICIT LOGICAL (L)
      LOGICAL INP
      COMMON /SDPARM/ LPDATA,LPBYTE,LPTRAK,LPSECT,LCBYTE,LCTRAK,LCSECT
C
C  Set for SemiDisk base I/O port address = 80H.
C
      LPDATA = 128
      LPBYTE = LPDATA + 1
      LPTRAK = LPDATA + 2
      LPSECT = LPDATA + 3
C
C  Set counters to zero and set SemiDisk
C
      LCBYTE=0
      LCTRAK=0
      LCSECT=0
      CALL SDSET
```

```
C
C  Clear Semidisk - Write zeros until track counter returns to zero
C
      IF(LCLR.NE.1) RETURN
      DO 3Ø IT=1, 256
      DO 2Ø IS=1, 16
      DO 1Ø IB=1, 128
      CALL SDWRIT(Ø)
   1Ø CONTINUE
   2Ø CONTINUE
   3Ø CONTINUE
C
C  Set counters to zero and set SemiDisk
C
      LCBYTE=Ø
      LCTRAK=Ø
      LCSECT=Ø
      CALL SDSET
C
      RETURN
      END


      SUBROUTINE SDSET
C  Set SemiDisk to desired byte, sector, and track
      IMPLICIT LOGICAL (L)
      LOGICAL INP
      COMMON /SDPARM/ LPDATA,LPBYTE,LPTRAK,LPSECT,LCBYTE,LCTRAK,LCSECT
      CALL OUT(LPBYTE,LCBYTE)
      CALL OUT(LPSECT,LCSECT)
      CALL OUT(LPTRAK,LCTRAK)
      RETURN
      END


      SUBROUTINE SDWRIT(LCHR)
C  Write one byte (LCHR) to SemiDisk and increment location
      IMPLICIT LOGICAL (L)
      LOGICAL INP
      COMMON /SDPARM/ LPDATA,LPBYTE,LPTRAK,LPSECT,LCBYTE,LCTRAK,LCSECT
C
C  Write byte into current location
C
      CALL OUT(LPDATA,LCHR)
C
C  Increment location if necessary
C
      CALL SDINCR
C
      RETURN
      END


      SUBROUTINE SDREAD(LCHR)
C  Read one byte (LCHR) to SemiDisk and increment location
      IMPLICIT LOGICAL (L)
      LOGICAL INP
      COMMON /SDPARM/ LPDATA,LPBYTE,LPTRAK,LPSECT,LCBYTE,LCTRAK,LCSECT
C
C  Read byte from current location
C
      LCHR=INP(LPDATA)
C
C  Increment location if necessary
C
      CALL SDINCR
C
      RETURN
      END
```

```
      SUBROUTINE SDINCR
C   Increment SemiDisk location pointers and set SemiDisk if necessary
      IMPLICIT LOGICAL (L)
      LOGICAL INP
      COMMON /SDPARM/ LPDATA,LPBYTE,LPTRAK,LPSECT,LCBYTE,LCTRAK,LCSECT
C
C   Next byte
C
      LCBYTE=LCBYTE + 1
      IF(LCBYTE.GT.Ø) RETURN
C
C   Next sector
C
      LCBYTE=Ø
      LCSECT=LCSECT + 1
      IF(LCSECT.LE.15) GO TO 1Ø
C
C   Next track
C
      LCSECT=Ø
      LCTRAK=LCTRAK+1
C
C   Set SemiDisk for new byte, sector, and track
C
   1Ø CALL SDSET
C
      RETURN
      END
```

A/D Operation

```
C   Program # FHR-345-, Ø9/Ø2/83
C   Four-channel A/D converter with output to SemiDisk
C
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      LOGICAL PEEK
      REAL V(4), VMAX, GAIN, GMAX, FREQ
      DIMENSION LBIT(8), LCBU(4), LCBL(4), LBIAS(4), LCHAN(4),
     >  LID(128), LR1(6), LR2(4)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
      COMMON /SDPARM/ LPDATA,LPBYTE,LPTRAK,LPSECT,LCBYTE,LCTRAK,LCSECT
C
      STOP=Ø
      WRITE(3,1Ø)
   1Ø FORMAT(' Four-channel A/D converter with SemiDisk output',
     >  ', Program # FHR-345-')
      CALL ADVERS(3)
C
C   Get A/D parameters
C
   2Ø WRITE(3,3Ø)
   3Ø FORMAT(/,' Physical channel numbers (Ø - 7) : ')
      DO 4Ø N = 1, 4
      CALL ATI(I,IERR)
      IF((IERR.NE.Ø) .OR. (I.LT.Ø) .OR. (I.GT.7)) GO TO 2Ø
      LCHAN(N) = I
   4Ø CONTINUE
C
      CALL ADGAIN(LGAIN)
C
      LSD = Ø
      LSR = Ø
C
C   Offset bias correction
C
      CALL OBC(LCHAN,LGAIN,LSD,LBIAS)
C
C   Form control bytes and addresses, set A/D converter
```

```
C
      DO 11Ø N=1, 4
      CALL ADSET(LCHAN(N),LGAIN,LSD,LBIAS(N),LCBU(N),LCBL(N))
  11Ø CONTINUE
C
C  Get run parameters
C
      WRITE(3,12Ø)
  12Ø FORMAT(/,' Number of 1ØØ-sample sets = ')
      CALL ATI(NSR,IERR)
      WRITE(3,13Ø)
  13Ø FORMAT(' Number of 3.3-ms intervals per sample = ')
      CALL ATI(NINT,IERR)
      NEXTRA = NINT - 2
      FREQ = 3ØØ.Ø / NINT
      WRITE(3,135) FREQ
  135 FORMAT(' Sampling frequency = ',F1Ø.3)
      WRITE(3,14Ø)
  14Ø FORMAT(/,' Recording ID = ')
      READ(3,15Ø) (LID(N), N=1,64)
  15Ø FORMAT(64A1)
C
C  Initialize semidisk
C
      CALL SDINIT(1)
C
C  Prepare identification data and write to SemiDisk
C
C  LID(1) - LID(64) are ASCII message from operator
      LID(65) = LGAIN
      CALL ILC(NSR,LID(66),LID(67))
      CALL ILC(NINT,LID(68),LID(69))
      DO 23Ø N = 7Ø, 128
      LID(N) = Ø
  23Ø CONTINUE
      DO 24Ø N = 1, 128
      CALL SDWRIT(LID(N))
  24Ø CONTINUE
C
C  Advise user to begin
C
      CALL OUTPUT(3,7)
      WRITE(3,3ØØ)
  3ØØ FORMAT(//,' NOW READY - Press RETURN to start')
      READ(3,31Ø)
  31Ø FORMAT()
      NSAMP = Ø
C
C  Begin taking measurements - reset clock
C
      CALL OUTPUT(3,7)
      CALL CLKSET
      DO 6ØØ NS = 1, NSR
      DO 5ØØ N1 = 1, 1ØØ
      CALL INTRPT(LINT)
      IF(LINT.EQ.27) GO TO 9ØØ
C
C  Digitize, then wait for end of first 3.3-ms interval
C
      DO 45Ø N=1, 4
      CALL ADCNV(LCBU(N),LCBL(N),LR2(N),LR1(N))
  45Ø CONTINUE
      CALL TIMER
C
C  Pack, then wait for end of second 3.3-ms interval
C
      CALL PACK(LR2,LR1)
      CALL TIMER
C
C  Transfer to SemiDisk, then wait for end of additional 3.3-m
      DO 48Ø N = 1, 6
```

```
      CALL SDWRIT(LR1(N))
  480 CONTINUE
      CALL WAIT(NEXTRA)
  500 CONTINUE
  600 CONTINUE
      CALL OUTPUT(3,7)
C
  900 CONTINUE
      STOP
      END


      SUBROUTINE OBC(LCHAN,LGAIN,LSD,LBIAS)
C  Offset bias correction
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      REAL V(4)
      DIMENSION LBIT(8), LCBU(4), LCBL(4), LBIAS(4), LCHAN(4)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
C
  100 WRITE(3,110)
  110 FORMAT(/,' Method for offset-bias correction ',
     >  '(1-assume, 2-measure, 3-disk) : ')
      CALL ATI(METHOD,IERR)
      IF((METHOD.LT.1).OR.(METHOD.GT.3)) GO TO 100
      IF(METHOD.EQ.2) GO TO 130
      IF(METHOD.EQ.3) GO TO 160
C
C  Assumed bias corrections
      DO 120 N=1, 4
      LBIAS(N) = 127
  120 CONTINUE
      GO TO 180
C
C  Measured bias corrections
  130 CONTINUE
      DO 140 N=1, 4
      CALL ADBIAS(LCHAN(N),LGAIN,LSD,LBIAS(N),V(N))
  140 CONTINUE
      WRITE(3,150) (V(N), N=1, 4)
  150 FORMAT(' Bias voltages = ',4F9.4)
      GO TO 180
C
C  Bias corrections from disk file
  160 CONTINUE
      CALL FOPENI(6,IERR)
      READ(6,170) (LBIAS(N), N=1, 4)
  170 FORMAT(4I5)
      ENDFILE 6
C
C  Display bias corrections
  180 WRITE(3,190) (LBIAS(N), N=1, 4)
  190 FORMAT(' Bias bytes =    ',4I9)
C
      RETURN
      END


      SUBROUTINE ILC(I,L1,L2)
C  Converts one integer to two bytes
C  L1 is high-order byte
C  Memory assignment for Microsoft FORTRAN-80
      IMPLICIT LOGICAL (L)
      LOGICAL PEEK
      M1=MLOC$(I)
      L2=PEEK(M1)
      M1=M1+1
      L1=PEEK(M1)
      RETURN
      END
```

```
      SUBROUTINE PACK(LR2,LR1)
C   Pack 8 bytes into 6 bytes
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      DIMENSION LBIT(8), LR1(1), LR2(1)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
C
      LR1(5) = LR2(1) .AND. (X'ØF')
      LR1(6) = LR2(3) .AND. (X'ØF')
C
      DO 1Ø I = 1, 4
      J = I + 4
      LTEST = LR2(2) .AND. LBIT(I)
      IF(LTEST.NE.Ø) LR1(5) = LR1(5) .OR. LBIT(J)
      LTEST = LR2(4) .AND. LBIT(I)
      IF(LTEST.NE.Ø) LR1(6) = LR1(6) .OR. LBIT(J)
   1Ø CONTINUE
C
      RETURN
      END


      SUBROUTINE CLKSET
C   Reset real-time clock in North-Star Horizon
      CALL OUT(6,X'5Ø')
      RETURN
      END


      SUBROUTINE WAIT(N)
C   Wait N time intervals
      DO 1Ø I=1, N
      CALL TIMER
   1Ø CONTINUE
      RETURN
      END


      SUBROUTINE TIMER
C   3.3-millisecond timer for North-Star Horizon
C   Note:  Jumper real-time clock for 3.3 ms
C   Resets clock at end of interval
C
      IMPLICIT LOGICAL (L)
      LOGICAL INP
C
C   Wait for clock flag
   1Ø L2=INP(6) .AND. 4
      IF(L2.EQ.Ø) GO TO 1Ø
C
C   Reset clock to start new interval
      CALL OUT(6,X'5Ø')
C
      RETURN
      END
```

SemiDisk to Floppy-Disk Transfer

```
C   Program # FHR-346A, Ø9/Ø2/83
C   SemiDisk to file transfer
C
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      LOGICAL PEEK
      REAL V(4), GAIN
      DIMENSION LBIT(8), LCBU(4), LCBL(4), LBIAS(4), LCHAN(4),
     >  LID(128), LR1(6), LR2(4)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
```

```
      COMMON /SDPARM/ LPDATA,LPBYTE,LPTRAK,LPSECT,LCBYTE,LCTRAK,LCSECT
C
      WRITE(3,10)
   10 FORMAT(' SemiDisk to file transfer,',
     >  ' Program # FHR-346A')
      CALL ADVERS(3)
      WRITE(3,20)
   20 FORMAT()
      CALL FOPENO(7,IERR)
      CALL SDINIT(0)
      WRITE(3,30)
   30 FORMAT(' Number of 100-sample sets = ')
      CALL ATI(NSR,IERR)
C
C  Identification data
C
      NREC = 1
      NLOC = 1
      LEND = 0
      DO 110 N = 1, 128
      CALL SDREAD(LCHR)
      CALL PUTCHR(NREC,NLOC,LCHR,LEND)
  110 CONTINUE
C
C  Measurements
C
      DO 600 NS = 1, NSR
      DO 500 N1 = 1, 100
C
C  Transfer six bytes
C
      DO 450 N=1, 6
      CALL SDREAD(LCHR)
      CALL PUTCHR(NREC,NLOC,LCHR,LEND)
  450 CONTINUE
C
  500 CONTINUE
  600 CONTINUE
C
C  Zero rest of file and close
C
      LEND = 1
      CALL PUTCHR(NREC,NLOC,LCHR,LEND)
      ENDFILE 7
      STOP
      END


      SUBROUTINE PUTCHR(NREC,NLOC,LCHR,LEND)
C  Unidirectional single-character write to disk file
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
C
      IF(LEND.NE.0) GO TO 10
C
      LBUF(NLOC) = LCHR
      NLOC = NLOC + 1
      IF(NLOC.LE.128) RETURN
C
      WRITE(7, REC=NREC) LBUF
      NLOC = 1
      NREC = NREC + 1
C
   10 CONTINUE
      DO 20 N = NLOC, 128
      LBUF(N) = 0
   20 CONTINUE
      WRITE(7, REC=NREC) LBUF
C
      RETURN
      END
```

Data Decoder

```
C  Program # FHR-347B, Ø9/Ø2/83
C  Decoder for A/D converter data
C
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      LOGICAL PEEK
      REAL V(4), GAIN, FREQ
      DIMENSION LBIT(8), LCBU(4), LCBL(4), LBIAS(4), LCHAN(4),
     >  LID(128), LR1(6), LR2(4)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
C
      WRITE(3,1Ø)
   1Ø FORMAT(' Decoder for A/D converter data',
     >  ', Program # FHR-347B')
      CALL ADVERS(3)
      WRITE(3,2Ø)
   2Ø FORMAT()
      CALL FOPENI(6,IERR)
C
C  Get identification data
C
      NREC = 1
      NLOC = 1
      DO 11Ø N = 1, 128
      CALL GETCHR(NREC,NLOC,LID(N),LEND)
      IF(LEND.NE.Ø) GO TO 9ØØ
  11Ø CONTINUE
C
      WRITE(3,12Ø) (LID(N), N=1, 64)
  12Ø FORMAT(/,' Data identification : ',64A1)
C
      LGAIN = LID(65)
      GAIN = Ø.Ø
      IF(LGAIN.EQ.Ø) GAIN = Ø.5
      IF(LGAIN.EQ.1) GAIN = 2.Ø
      IF(LGAIN.EQ.2) GAIN = 8.Ø
      IF(LGAIN.EQ.3) GAIN = 32.Ø
      WRITE(3,13Ø) GAIN, LGAIN
  13Ø FORMAT(' Gain, gain code = ',F5.2,I5)
C
      CALL LIC(LID(66),LID(67),NSR)
      WRITE(3,14Ø) NSR
  14Ø FORMAT(' Number of 1ØØ-sample sets = ',I8)
C
      CALL LIC(LID(68),LID(69),NINT)
      WRITE(3,15Ø) NINT
  15Ø FORMAT(' Number of 3.3-ms intervals / sample = ',I8)
      IF(NINT.EQ.Ø) GO TO 17Ø
      FREQ = 3ØØ.Ø / NINT
      WRITE(3,16Ø) FREQ
  16Ø FORMAT(' Sampling frequency = ',F1Ø.3)
  17Ø CONTINUE
C
C  Measurements
C
      WRITE(3,3ØØ)
  3ØØ FORMAT(/,'   Nset',4X,'N1',6X,'V1',7X,'V2',7X,'V3',
     >  7X,'V4',/)
      DO 6ØØ NS = 1, NSR
      DO 5ØØ N1 = 1, 1ØØ
C
C  Get six bytes from disk
C
      DO 45Ø N=1, 6
      CALL GETCHR(NREC,NLOC,LR1(N),LEND)
      IF(LEND.NE.Ø) GO TO 9ØØ
```

```
  45Ø CONTINUE
C
C  Unpack and print
C
      CALL UNPACK(LR2,LR1)
      DO 46Ø N=1, 4
      CALL ADVOLT(LR2(N),LR1(N),LGAIN,LCHD,V(N))
  46Ø CONTINUE
      WRITE(3,47Ø) NS, N1, (V(N), N=1, 4)
  47Ø FORMAT(' ',2I6,4F9.3)
C
  5ØØ CONTINUE
  6ØØ CONTINUE
C
  9ØØ WRITE(3,91Ø)
  91Ø FORMAT(/,' EOF encountered',/)
      ENDFILE 6
      STOP
      END


      SUBROUTINE UNPACK(LR2,LR1)
C  Unpack 6 bytes into 8 bytes
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      DIMENSION LBIT(8), LR1(1), LR2(1)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
C
      LR2(1) = LR1(5) .AND. (X'ØF')
      LR2(3) = LR1(6) .AND. (X'ØF')
C
      LR2(2) = Ø
      LR2(4) = Ø
      DO 1Ø I = 1, 4
      J = I + 4
      LTEST = LR1(5) .AND. LBIT(J)
      IF(LTEST.NE.Ø) LR2(2) = LR2(2) .OR. LBIT(I)
      LTEST = LR1(6) .AND. LBIT(J)
      IF(LTEST.NE.Ø) LR2(4) = LR2(4) .OR. LBIT(I)
   1Ø CONTINUE
C
      RETURN
      END


      SUBROUTINE LIC(L1,L2,I)
C  Converts two bytes to one integer
C  L1 is high-order byte
C  Memory assignment for Microsoft FORTRAN-8Ø
      IMPLICIT LOGICAL (L)
      M1=MLOC$(I)
      CALL POKE(M1,L2)
      M1=M1+1
      CALL POKE(M1,L1)
      RETURN
      END


      SUBROUTINE GETCHR(NREC,NLOC,LCHR,LEND)
C  Unidirectional single-character read from disk file
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
C
      IF(NLOC.EQ.1) READ(6, REC=NREC, END=1Ø) LBUF
C
      LCHR = LBUF(NLOC)
      LEND = Ø
      NLOC = NLOC + 1
      IF(NLOC.LE.128) RETURN
C
      NLOC = 1
```

```
      NREC = NREC + 1
      RETURN
C
   10 LCHR = Ø
      LEND = 1
      RETURN
      END
```

APPENDIX B. COMMUNICATION PROGRAM

```
C  Program # FHR-350B, 09/27/83
C  Transmission of binary files to HP-1000 via RS-232
C  HP binary-block protocol:
C  Send:
C     STX (start block), SUB (EOF), or EOT (all done)
C     128 8-bit binary data bytes
C     Check byte
C  Answer:
C     ACK or NAK
C  Repeat up to two times if NAK received
C     Intersperse HP protocol for HP terminals
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF
      DATA LREAD /-1/
      DATA LUNI, LUNO /0,0/
      DATA LISP,LISM,LIDP,LIDM /0,0,0,0/
      DATA LOSP,LOSM,LODP,LODM /0,0,0,0/
      DATA NEMAX /3/
      CALL GETLUN(LUNC,LUND)
C
      WRITE(LUNC,10)
   10 FORMAT(' HP binary-block transmitter, Program # FHR-350B',/,
     >  ' Copyright (C) 1983 by Green Mountain Radio Research',/)
C
C  Get I/O parameters
C
      IF(LREAD.NE.-1) GO TO 290
      LREAD=1
C  Input channel
  100 WRITE(LUNC,110)
  110 FORMAT(' Input channel :')
      CALL GETPRT(LUNC,LISP,LISM,LIDP,LIDM)
C  Output channel
  200 WRITE(LUNC,210)
  210 FORMAT(' Output channel : ')
      CALL GETPRT(LUNC,LOSP,LOSM,LODP,LODM)
C
C  Clear I/O port
C
  290 WRITE(LUNC,310)
  310 FORMAT(/,' Press RETURN when ready')
      CALL OUTPUT(LUNC,7)
      READ(LUNC,320)
  320 FORMAT()
      CALL CLRIO
C
C  Begin transfer - open disk file
C
  300 CALL OUTPUT(LUNC,7)
      CALL FOPENI(LUND,IERR)
      IREC=1
C
C  Get disk record and transmit
C
  400 CONTINUE
C
C  Wait for HP to send DC1
C
      CALL GETDC1
C
C  Read record
C
      READ(LUND, REC=IREC, END=1000) LBUF
      NERROR = 0
```

```
  410 WRITE(3,420) IREC
  420 FORMAT(' # ',I6)
      CALL SNDREC
      CALL HP1
      CALL GETACK(LERROR)
      CALL HP2
      IF(LERROR.EQ.0) GO TO 490
      CALL OUTPUT(LUNC,7)
      NERROR = NERROR + 1
      IF(NERROR.LT.NEMAX) GO TO 410
      GO TO 1000
C
  490 IREC=IREC+1
      GO TO 400
C
C  Done - close file and see whether to continue
C
 1000 ENDFILE LUND
      WRITE(LUNC,1010)
 1010 FORMAT(/,' Continue ? ')
      READ(3,1020) LANS
 1020 FORMAT(A1)
      LANS=LUPPER(LANS)
      IF(LANS.NE.89) GO TO 1100
C
C  End of file but more files - send SUB, then set of null
      CALL XCZ
      GO TO 300
C
C  No more files - send EOT and set of nulls
 1100 CALL XEOT
      CALL HALT
      STOP
      END


      SUBROUTINE GETLUN(LUNC,LUND)
C  Get console and disk LUN
      IMPLICIT LOGICAL (L)
      LUNC=3
      LUND=6
      RETURN
      END


      SUBROUTINE GETPRT(LUNC,LISP,LISM,LIDP,LIDM)
C  Get I/O port parameters
      IMPLICIT LOGICAL (L)
   10 FORMAT(2A1)
C
   20 WRITE(LUNC,30)
   30 FORMAT(' Status port (2-digit hex) = ')
      READ(LUNC,10) LH2,LH1
      LH4='0'
      LH3='0'
      CALL HIC(LH4,LH3,LH2,LH1,I,LERR)
      IF(LERR.NE.0) GO TO 20
      LISP=I
C
   40 WRITE(LUNC,50)
   50 FORMAT(' Status mask (2-digit hex) = ')
      READ(LUNC,10) LH2,LH1
      LH4='0'
      LH3='0'
      CALL HIC(LH4,LH3,LH2,LH1,I,LERR)
      IF(LERR.NE.0) GO TO 40
      LISM=I
C
   60 WRITE(LUNC,70)
   70 FORMAT(' Data port (2-digit hex) = ')
      READ(LUNC,10) LH2,LH1
```

```
      LH4='Ø'
      LH3='Ø'
      CALL HIC(LH4,LH3,LH2,LH1,I,LERR)
      IF(LERR.NE.Ø) GO TO 6Ø
      LIDP=I
C
   8Ø WRITE(LUNC,9Ø)
   9Ø FORMAT(' Data mask (2-digit hex) = ')
      READ(LUNC,1Ø) LH2,LH1
      LH4='Ø'
      LH3='Ø'
      CALL HIC(LH4,LH3,LH2,LH1,I,LERR)
      IF(LERR.NE.Ø) GO TO 8Ø
      LIDM=I
C
      RETURN
      END


      SUBROUTINE SNDREC
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF
      DATA LSTX / 2 /
C
C  Send start-transmission character
C
      CALL OUTPUP(LSTX)
C
C  Send 128 data bytes
C
      DO 1ØØ I=1, 128
      CALL OUTPUP(LBUF(I))
  1ØØ CONTINUE
C
C  Compute and send parity check
C
      LPAR = Ø
      DO 2ØØ I = 1, 128
      LPAR = LPAR .XOR. LBUF(I)
  2ØØ CONTINUE
      CALL OUTPUP(LPAR)
C
      RETURN
      END


      SUBROUTINE XCZ
C  Transmit control-Z to signal end of transmission
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF
C
      CALL OUTPUP(26)
      DO 1Ø I = 1, 129
      CALL OUTPUP(Ø)
   1Ø CONTINUE
C
      CALL HP1
C
      RETURN
      END


      SUBROUTINE XEOT
C  Transmit control-D to signal all finished
      IMPLICIT INTEGER (A-K,M-Z)
```

```
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF
C
      CALL OUTPUP(4)
      DO 1Ø I = 1, 129
      CALL OUTPUP(Ø)
   1Ø CONTINUE
C
      CALL HP1
C
      RETURN
      END


      SUBROUTINE GETACK(LERROR)
C  Get acknowledgement from receiver
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF
      DATA LACK /6/
C
      CALL INPUP(LRCV)
      LERROR = Ø
      IF(LRCV.EQ.LACK) RETURN
      LERROR = 1
C
      RETURN
      END


      SUBROUTINE INPUP(LCHR)
C  Direct input from computer I/O port
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF
C
C  Check status, wait for something to happen
  1ØØ LCHR=INP(LISP)
      LCHR=LCHR.AND.LISM
      IF(LCHR.EQ.Ø) GO TO 1ØØ
C
C  Something came in; get it
      LCHR=INP(LIDP)
C
      RETURN
      END


      SUBROUTINE OUTPUP(LCHR)
C  Direct output to computer I/O port of 8Ø8Ø computer
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128), LBIT(8)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF
      DATA LBIT /1, 2, 4, 8, 16, 32, 64, X'8Ø'/
C
C  Check status - wait until OK
C
   1Ø LSTAT=INP(LOSP) .AND. LOSM
      IF(LSTAT.EQ.Ø) GO TO 1Ø
C
C  Send character
C
      CALL OUT(LODP,LCHR)
```

```
      RETURN
      END


      SUBROUTINE CLRIO
C Clear/initialize I/O ports
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF

      LCHR=INP(LIDP)
      RETURN
      END


      SUBROUTINE GETDC1
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF
      DATA LDC1 /17/

      CALL INPUP(LCHR)
      IF(LCHR.EQ.LDC1) RETURN
      WRITE(LUNC,1Ø)
 1Ø   FORMAT(' *** FIRST CHAR NOT DC1 ***')

      CALL HALT
      RETURN
      END


      SUBROUTINE HP1
C HP-1ØØØ transmission terminator
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF

C Receive ENQ and send ACK (twice)

      CALL ENQACK
      CALL ENQACK

      RETURN
      END


      SUBROUTINE HP2
C HP-1ØØØ reception terminator
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF

C Get CR LF

      CALL INPUP(LCHR)
      CALL INPUP(LCHR)

C Receive ENQ and send ACK (twice)

      CALL ENQACK
      CALL ENQACK
      CALL ENQACK
```

```
C
      RETURN
      END


      SUBROUTINE ENQACK
C  Receive ENQ, send ACK
      IMPLICIT INTEGER (A-K,M-Z)
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /IO/ LUNC,LUNI,LUNO,LPAR,LISP,LISM,LIDP,LIDM,
     >  LOSP,LOSM,LODP,LODM,LBUF
      DATA LENQ, LACK /5, 6/
C
      CALL INPUP(LCHR)
      IF(LCHR.NE.LENQ) GO TO 1ØØ
      CALL OUTPUP(LACK)
      RETURN
C
C  Error
  1ØØ WRITE(LUNC,11Ø) LCHR
  11Ø FORMAT(' *** ENQ NOT RECEIVED ***',I5)
      CALL HALT
      RETURN
      END
```

APPENDIX C. PROCESSING PROGRAM

Digitized-Noise Simulator

```
C  Program # FHR-352A, 1Ø/28/83
C  ELF-noise generator with disk-file output
C
      IMPLICIT LOGICAL (L)
      DIMENSION P(4),GAMMAL(4),A(4),CHX(4),CHY(4),PSI(4),R(4),
     >  SIGX(4), HXOLD(1ØØ), HYOLD(1ØØ), FW(1ØØ),
     >  ZW(3), YW(3), ZR(3), ZI(3), YR(3), YI(3),
     >  LR1(6), LR2(4), LID(128)
      COMMON /BLOCK1/ IR2,IR1,P,A,CHX,CHY,PSI,R,SIGX,PSIMIN,PSID
      COMMON /BLOCK2/ NFILT, FW
      COMMON /PCPARM/ NREC, NBYTE
      DATA PSI(1),PSI(2),PSI(3),PSI(4) /45.Ø,135.Ø,-7Ø.Ø,Ø.Ø/
      DATA LGAIN, LCHAN /1,Ø/
   1Ø FORMAT(' ELF-noise generator with disk-file output, ',
     >  ' Program # FHR-352A',/)
C
C  Fixed parameters and constants
C
      DR=Ø.Ø17453293
      TWOPI=6.2831852
      SIGMAG=1.Ø
      KMOD=1ØØ
      FSAMP=1ØØ.Ø
      CFILT=1.Ø
      FSIG=1Ø.Ø
      ILS=1
      ILY=3
      ILY2=ILY*2
      N3MS=3
      NREC=1
      NBYTE=1
C
C  Announce name and get other parameters
C
      WRITE(3,1Ø)
      CALL ADVERS(3)
      CALL ADSET(LCHAN,LGAIN,LSD,LBIAS,LCBU,LCBL)
   9Ø WRITE(3,1ØØ)
  1ØØ FORMAT(/,' INPUT:  Noise-source parameters',/,
     >  '  #',2X,'P, Gamma, A, Psimin, Psimax (degrees)')
      DO 12Ø I=1, 4
      WRITE(3,11Ø) I
  11Ø FORMAT(' ',I2,2X)
      CALL ATR(P(I),IERR)
      IF(IERR.EQ.27) GO TO 9Ø
      CALL ATR(GAMMAL(I),IERR)
      IF(IERR.EQ.27) GO TO 9Ø
      CALL ATR(A(I),IERR)
      IF(IERR.EQ.27) GO TO 9Ø
  12Ø CONTINUE
      CALL ATR(PSIMIN,IERR)
      IF(IERR.EQ.27) GO TO 9Ø
      CALL ATR(PSIMAX,IERR)
      IF(IERR.EQ.27) GO TO 9Ø
      PSID=PSIMAX-PSIMIN
C
C  Type of reference
      WRITE(3,13Ø)
  13Ø FORMAT(/,' Reference (E or M) ? ')
      READ(3,135) LREF
  135 FORMAT(A1)
      LREF=LUPPER(LREF)
C
C  Local noise level
      WRITE(3,14Ø)
```

```
  140 FORMAT(' Plocal = ')
      CALL ATR(PLOCAL,IERR)
      SIGLCL=SQRT(PLOCAL)
C
C  Primary-antenna channel
      WRITE(3,155)
  155 FORMAT(' Primary-antenna axis (X, Y, or Z) = ')
      READ(3,156) LPCH
      LPCH = LUPPER(LPCH)
  156 FORMAT(A1)
      IPCH = LPCH - 87
C
C  Other parameters
      WRITE(3,200)
  200 FORMAT(/,' Nsec = ')
      CALL ATI(NSEC,IERR)
      WRITE(3,220)
  220 FORMAT(' Nfilter, Yzx, Yzy, Yzz = ')
      CALL ATI(NFILT,IERR)
      CALL ATR(YZX,IERR)
      CALL ATR(YZY,IERR)
      CALL ATR(YZZ,IERR)
      WRITE(3,240)
  240 FORMAT(' Seeds = ')
      CALL ATI(ISEED1,IERR)
      CALL ATI(ISEED2,IERR)
      IF((ISEED1.NE.0).AND.(ISEED2.NE.0)) GO TO 250
      CALL SEED(ISEED1)
      CALL SEED(ISEED2)
  250 IR1=ISEED1
      IR2=ISEED2
      WRITE(3,260)
  260 FORMAT(' Output voltages ? ')
      READ(3,156) LOV
      LOV = LUPPER(LOV)
C
C  Initialize output file and write ID to it
C
      CALL FOPENO(6,IERR)
      CALL IDPREP(LGAIN,NSEC,N3MS,LID)
      DO 270 I=1, 128
      CALL PUTCHR(LID(I))
  270 CONTINUE
C
C  Write parameters and headings
C
      WRITE(3,300)
  300 FORMAT(23X,'Noise-Source Data',/,'   I',5X,'P',9X,'Gamma',9X,
     >  'A',6X,'Psi/Psimin',2X,'Psimax',/)
      DO 340 I=1,4
      IF(I.EQ.4) GO TO 320
      WRITE(3,310) I, P(I), GAMMAL(I), A(I), PSI(I)
  310 FORMAT(' #',I2,3G12.4,F8.1,2X,F8.1)
      GO TO 330
  320 WRITE(3,310) I, P(I), GAMMAL(I), A(I), PSIMIN, PSIMAX
  330 CONTINUE
      PSI(I)=PSI(I)*DR
      CHX(I)=-SIN(PSI(I))
      CHY(I)=COS(PSI(I))
  340 CONTINUE
      PSIMIN=PSIMIN*DR
      PSIMAX=PSIMAX*DR
      PSID=PSID*DR
      PT=0.0
      DO 345 I=1,4
      PT=PT+P(I)
  345 CONTINUE
      PT4=PT*4.0
      PT9=PT*9.0
      WRITE(3,346) PT
  346 FORMAT(' T  ',G12.4)
```

```
C
C  Output headings
C
      IF(LOV.EQ.89) WRITE(3,4Ø5)
  4Ø5 FORMAT(/,2X,'Nsec',2X,'N1Øms',6X,'Vxr',6X,'Vyr',
     >  6X,'Vzr',6X,'Vp',/)
C
C  Initialize parameters
C
      ISEED1=IR1
      ISEED2=IR2
      CALL RANDU(IR2,IR1)
      CALL RANDU(IR2,IR1)
      DO 41Ø I=1,4
      IF(P(I).GT.Ø.Ø) CALL ELFPRM(P(I),GAMMAL(I),A(I),SIGX(I),R(I))
  41Ø CONTINUE
      DO 44Ø J=1,NFILT
      HXOLD(J)=Ø.Ø
      HYOLD(J)=Ø.Ø
      FW(J)=PR(CFILT,J)
  44Ø CONTINUE
C
C  Begin seconds loop
C
      DO 7ØØØ KN=1, NSEC
      WRITE(3,49Ø) KN
  49Ø FORMAT(' T = ',I5)
C
C  Begin 1Ø-ms loop
C
      DO 1ØØØ KM = 1, KMOD
C
      CALL INTRPT(LINT)
      IF(LINT.EQ.27) GO TO 9999
C
C  Begin noise generation
C
C  Generate incident noise fields
      CALL ELFINC(ZW(1),ZW(2),EZ)
C
C  Horizontal electric fields
      YW(1)=FILTER(ZW(2),HYOLD)
      YW(2)=FILTER(ZW(1),HXOLD)
C
C  Vertical magnetic field
      ZW(3)=YZX*ZW(1) + YZY*ZW(2) + YZZ*EZ
C
C  Vertical electric field
      YW(3)=EZ
C
C  Magnetic reference antenna?
      IF(LREF.EQ.69) GO TO 51Ø
      DO 5ØØ I=1,3
      YW(I)=ZW(I)
  5ØØ CONTINUE
  51Ø CONTINUE
C
C  Generate and add local noise to reference-antenna output
C
      DO 52Ø I=1, 3
      CALL GAUSS(IR2,IR1,Ø.Ø,SIGLCL,GN)
      YW(I)=YW(I) + GN
  52Ø CONTINUE
C
C  Select primary channel
      ZW(1) = ZW(IPCH)
C
C  Quantize
      DO 53Ø I=1, 3
      CALL ADSIM(LGAIN,LCHAN,YW(I),LR2(I), LR1(I))
  53Ø CONTINUE
```

```
      CALL ADSIM(LGAIN,LCHAN,ZW(1),LR2(4), LR1(4))
      DO 58Ø I = 1, 3
      CALL ADVOLT(LR2(I),LR1(I),LGAIN,LCHANO,YW(I))
  58Ø CONTINUE
      CALL ADVOLT(LR2(4),LR1(4),LGAIN,LCHANO,ZW(1))
C
C  Pack and output
      IF(LOV.EQ.89) WRITE(3,7ØØ) KN, KM, (YW(I), I=1,3), ZW(1)
  7ØØ FORMAT(' ',2I6,4F9.3)
      CALL PACK(LR2,LR1)
      DO 71Ø I=1, 6
      CALL PUTCHR(LR1(I))
  71Ø CONTINUE
C
C  End 1Ø-ms loop
C
  9ØØ CONTINUE
 1ØØØ CONTINUE
C
 699Ø CONTINUE
 7ØØØ CONTINUE
C
 9ØØØ CONTINUE
C
C
 9999 CALL PCEND
      ENDFILE 6
      STOP
      END


      SUBROUTINE ELFINC(HX,HY,EZ)
C  INCIDENT ELF FIELDS
      DIMENSION P(4),A(4),CHX(4),CHY(4),PSI(4),R(4),
     >  SIGX(4)
      COMMON /BLOCK1/ IR2,IR1,P,A,CHX,CHY,PSI,R,SIGX,PSIMIN,PSID
  1ØØ HX=Ø.Ø
      HY=Ø.Ø
      EZ=Ø.Ø
      DO 31Ø I=1,4
      IF(P(I).LE.Ø.Ø) GO TO 3ØØ
      IF(I.LT.4) GO TO 2ØØ
C  GET RANDOM AZIMUTH ANGLE FOR SOURCE #4
      CALL RANDU(IR2,IR1)
      IØ=IABS(IR1)
      PSI(4)=(IØ/32767.Ø)*PSID + PSIMIN
      CHX(4)=-SIN(PSI(4))
      CHY(4)=COS(PSI(4))
C
C  COMPUTE, DECOMPOSE, AND SUM OUTPUTS FROM NOISE SOURCES
C  ELF noise via FLM model
  2ØØ CALL GAUSS(IR2,IR1,Ø.Ø,SIGX(I),Z)
      CALL PWRRAY(IR2,IR1,R(I),A(I),Y)
      H=Z+Y
C  Incident fields
      HX=HX+H*CHX(I)
      HY=HY+H*CHY(I)
      EZ=EZ+H
  3ØØ CONTINUE
  31Ø CONTINUE
      RETURN
      END


      SUBROUTINE ELFPRM(PZ,GAMMAL,A,SIGMAX,R)
C  CALCULATION OF PARAMETERS FOR SUBROUTINE ELFNSE
C  FOR GAUSSIAN COMPONENT
C  GAMMAL=IMPUSLIVITY; GAMMA=GAMMA FUNCTION
      PX=PZ/(1.Ø + GAMMAL**2)
      SIGMAX=SQRT(PX)
C  FOR POWER-RAYLEIGH COMPONENT
```

```
      R=SQRT((PZ-PX)/GAMMA(1.Ø + 2.Ø/A))
      RETURN
      END


      FUNCTION FILTER(H,HOLD)
      DIMENSION HOLD(1ØØ), FW(1ØØ)
      COMMON /BLOCK2/ NFILT, FW
C  SAVE H
      DO 1Ø J=2,NFILT
      JM=J-1
      HOLD(J)=HOLD(JM)
   1Ø CONTINUE
      HOLD(1)=H
C  COMPUTE RESPONSE
      FILTER=Ø.Ø
      DO 2Ø J=1,NFILT
      FILTER=FILTER + HOLD(J)*FW(J)
   2Ø CONTINUE
      RETURN
      END


      FUNCTION PR(C,K)
C  PULSE RESPONSE OF FLAT, HOMOGENEOUS CONDUCTING GROUND
      IF(K.GE.2) GO TO 2ØØ
      IF(K.EQ.1) GO TO 1ØØ
C  K <= Ø
      PR=Ø.Ø
      RETURN
C  K = 1
  1ØØ PR=C
      RETURN
C  K >= 2
  2ØØ PR=C*(1.Ø/(1.Ø*K)**Ø.5 - 1.Ø/(1.Ø*K-1.Ø)**Ø.5)
      RETURN
      END


      SUBROUTINE ILC(I,L1,L2)
C  Converts one integer to two bytes
C  L1 is high-order byte
C  Memory assignment for Microsoft FORTRAN-8Ø
      IMPLICIT LOGICAL (L)
      LOGICAL PEEK
      M1=MLOC$(I)
      L2=PEEK(M1)
      M1=M1+1
      L1=PEEK(M1)
      RETURN
      END


      SUBROUTINE PACK(LR2,LR1)
C  Pack 8 bytes into 6 bytes
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      DIMENSION LBIT(8), LR1(1), LR2(1)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
C
      LR1(5) = LR2(1) .AND. (X'ØF')
      LR1(6) = LR2(3) .AND. (X'ØF')
C
      DO 1Ø I = 1, 4
      J = I + 4
      LTEST = LR2(2) .AND. LBIT(I)
      IF(LTEST.NE.Ø) LR1(5) = LR1(5) .OR. LBIT(J)
      LTEST = LR2(4) .AND. LBIT(I)
      IF(LTEST.NE.Ø) LR1(6) = LR1(6) .OR. LBIT(J)
   1Ø CONTINUE
C
```

```
      RETURN
      END


      SUBROUTINE IDPREP(LGAIN,NSR,NINT,LID)
C  Prepare identification data
      IMPLICIT LOGICAL (L)
      DIMENSION LID(128)
C
C  LID(1) - LID(64) are ASCII message from operator
      WRITE(3,140)
  140 FORMAT(/,' Recording ID = ')
      READ(3,150) (LID(N), N=1,64)
  150 FORMAT(64A1)
C
C  Gain (LGAIN), number of seconds (NSR), number of samples/second (NINT)
      LID(65) = LGAIN
      CALL ILC(NSR,LID(66),LID(67))
      CALL ILC(NINT,LID(68),LID(69))
C
      DO 230 N = 70, 128
      LID(N) = 0
  230 CONTINUE
C
      RETURN
      END


      SUBROUTINE PUTCHR(LCHR)
C  Put single character to disk file
C  Must be sequential
C  Set NREC = NBYTE = 1 to start
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
      COMMON /PCPARM/ NREC, NBYTE
C
C  Put character in buffer
C
      LBUF(NBYTE) = LCHR
      NBYTE = NBYTE + 1
      IF(NBYTE.NE.129) RETURN
C
C  Put buffer on disk
C
      WRITE(6, REC=NREC) LBUF
      NBYTE = 1
      NREC = NREC + 1
C
      RETURN
      END


      SUBROUTINE PCEND
C  Dump buffer for PUTCHR
      IMPLICIT LOGICAL (L)
      COMMON /PCPARM/ NREC, NBYTE
C
   10 IF(NBYTE.EQ.1) GO TO 20
      CALL PUTCHR(0)
      GO TO 10
C
   20 RETURN
      END
```

Digitized-Noise Processor

```
C  Program # FHR-353E, 11/11/83
C  Single-channel NLP/ANC algorithm with disk-file input
C
      IMPLICIT LOGICAL (L)
      DIMENSION
     >  SR(3), SI(3), ZW(3), YW(3), ZR(3), ZI(3), YR(3), YI(3),
     >  PHATZ(3), PHATY(3), NNLZ(3), NNLY(3),
     >  ZBARR(3), ZBARI(3), YBARR(3), YBARI(3),
     >  SIGZR(3), SIGZI(3), SIGYR(3), SIGYI(3),
     >  SIGYYR(3,3), SIGYYI(3,3), SIGZYR(3,3), SIGZYI(3,3),
     >  RYYR(3,3), RYYI(3,3), RZYR(3,3), RZYI(3,3),
     >  CONJ(3), DUM1(3,3), DUM2(3,3), DUM3(3,3),
     >  SHATR(3), SHATI(3), RXYR(3,3), RXYI(3,3),
     >  DUM4(6,6), DUM5(6,6), DUM6(6,6), RINVR(3,3), RINVI(3,3),
     >  WR(3,3), WI(3,3),ZRL(3),YRL(3),ZIL(3),YIL(3),
     >  SIGZRL(3),  SIGYRL(3), SIGZIL(3), SIGYIL(3),
     >  ZBARRL(3), ZBARIL(3), YBARRL(3), YBARIL(3),
     >  LR1(6), LR2(4), LID(128), LBIT(8), CAL(4),
     >  ZRF(3), ZIF(3), ZBARRF(3), ZBARIF(3), SIGZRF(3), SIGZIF(3)
      COMMON /FLTR/ FGAIN, FTHR, TOTSMP, COSWAV, SINWAV
      COMMON /ADA/ LBIT, LSR, IABASE, IACBU, IACBL
      DATA LGAIN, LCHAN /1,Ø/
   1Ø FORMAT(' Single-channel NLP/ANC algorithm with disk-',
     > 'file input',/,' Program # FHR-353E',/)
C
C  Fixed parameters and constants
C
      DR=Ø.Ø17453293
      TWOPI=6.2831852
      SIGMAG=1.Ø
      KMOD=1ØØ
      FSAMP=1ØØ.Ø
      CFILT=1.Ø
      FSIG=1Ø.Ø
      ILS=1
      ILY=3
      ILY2=ILY*2
C
C  Announce name, get number of disk files to process
C
      WRITE(3,1Ø)
      CALL ADVERS(3)
      WRITE(3,2Ø)
   2Ø FORMAT()
      CALL ADSET(LCHAN,LGAIN,LSD,LBIAS,LCBU,LCBL)
C
C  Get number of files to process, open files
C
      CALL FOPENO(1Ø,IERR)
   25 WRITE(3,3Ø)
   3Ø FORMAT(/,' Number of disk files to process = ')
      CALL ATI(NDMAX,IERR)
      IF((NDMAX.LT.1).OR.(NDMAX.GT.4)) GO TO 25
      DO 4Ø ND=1, NDMAX
      LUN = ND + 5
      CALL FOPENI(LUN,IERR)
   4Ø CONTINUE
C
C  Start disk file
C
      DO 95ØØ ND = 1, NDMAX
      LUN = ND + 5
      NREC=1
      NLOC=1
      WRITE(3,5Ø) ND
      WRITE(1Ø,5Ø) ND
   5Ø FORMAT(/,' ',8Ø('='),/,' Disk # ',I1,/,)
C
C  Get data ID, get calibration constants
```

```
C
      DO 8Ø I=1, 128
      CALL GETCHR(LUN,NREC,NLOC,LID(I),LEND)
      IF(LEND.NE.Ø) GO TO 85
   8Ø CONTINUE
      CALL GETIDD(LID,LGAIN,GAIN,NSR,NINT,FREQ)
      CALL GETCAL(LUN,NREC,NLOC,ND,LGAIN,CAL,NSC,NSS,LERR)
      IF(LERR.NE.Ø) GO TO 85
      GO TO  9Ø
   85 WRITE(3,86)
   86 FORMAT(' *** BAD FORMAT - INPUT FILE ***')
      GO TO 9999
C
   9Ø CONTINUE
C
C  Signal parameters
      IF(ND.NE.1) GO TO 3ØØ
      WRITE(3,16Ø)
  16Ø FORMAT(/,' Signal components (I, Q) = ')
      CALL ATR(SR(1),IERR)
      CALL ATR(SI(1),IERR)
      WRITE(3,17Ø)
  17Ø FORMAT(' Axis (X, Y, or Z) for fixed cancellation : ')
      READ(3,18Ø) LAXIS
  18Ø FORMAT(A1)
      LAXIS = LUPPER(LAXIS)
      IAXIS = LAXIS - 87
C
C  Other parameters
      WRITE(3,2ØØ)
  2ØØ FORMAT(' Nsec, Kout = ')
      CALL ATI(NSEC,IERR)
      CALL ATI(KOUT,IERR)
      WRITE(3,21Ø)
  21Ø FORMAT(' Nset, F3dB, Thresh = ')
      CALL ATI(NSET,IERR)
      CALL ATR(F3DB,IERR)
      CALL ATR(FTHR,IERR)
      KMAX=NSEC+NSET
      KAVG=NSET + NSEC/2
      WRITE(3,23Ø)
  23Ø FORMAT(' Number of runs = ')
      CALL ATI(NRUN,IERR)
C
C  Write parameters and headings to output files
C
  3ØØ WRITE(1Ø,1Ø)
C
      WRITE(1Ø,3Ø5) (LID(I), I=1,64)
  3Ø5 FORMAT(' Noise-file parameters:',//,' ID: "',64A1,'"')
      WRITE(1Ø,31Ø) GAIN, FREQ
  31Ø FORMAT(' A/D gain = ',F5.1,'  Sampling frequency = ',F5.1)
      WRITE(1Ø,32Ø) NSR
  32Ø FORMAT(' Total seconds in noise file = ',I5)
      WRITE(1Ø,33Ø) (CAL(I), I=1,4)
  33Ø FORMAT(' 1-V Calibration levels = ',4F9.3)
      WRITE(1Ø,34Ø) NSC, NSS
  34Ø FORMAT(' Calibration, start times (s) = ',2I8)
C
      WRITE(1Ø,37Ø) SR(1), SI(1)
  37Ø FORMAT(/,' Processing parameters:',//,
     >  ' Signal = ', G12.4, ' +j ', G12.4)
      WRITE(1Ø,38Ø) LAXIS
  38Ø FORMAT(' Axis for fixed cancellation: ',A1)
      WRITE(1Ø,39Ø) NSEC, KOUT
  39Ø FORMAT(' Nsec = ',I6,' Kout = ',I6)
      WRITE(1Ø,391) NSET, F3DB, FTHR
  391 FORMAT(' Nset = ',I6,' F3dB = ',F5.2, '  Thresh = ',G12.4)
C
C  Start run
C
```

```
      DO 9000 NR=1, NRUN
      WRITE(3,400) NR
      WRITE(10,400) NR
  400 FORMAT(/,' ',80('-'),/,' Run # ',I3)
C
C  Output headings
C
      WRITE(3,405)
      WRITE(10,405)
  405 FORMAT(/,4X,'T',6X,'MSEL',9X,'Gn',6X,'Ga',6X,'Gta',
     >  5X,'Gf',6X,'Gtf',/)
C
C  Initialize parameters
C
      DO 430 J=1,3
      ZRL(J)=0.0
      ZIL(J)=0.0
      YRL(J)=0.0
      YIL(J)=0.0
      ZR(J)=0.0
      ZI(J)=0.0
      YR(J)=0.0
      YI(J)=0.0
      SIGZR(J)=0.0
      SIGZI(J)=0.0
      SIGYR(J)=0.0
      SIGYI(J)=0.0
      SIGZRL(J)=0.0
      SIGZIL(J)=0.0
      SIGYRL(J)=0.0
      SIGYIL(J)=0.0
      SIGZRF(J)=0.0
      SIGZIF(J)=0.0
      DO 420 I=1,3
      SIGYYR(I,J)=0.0
      SIGYYI(I,J)=0.0
      SIGZYR(I,J)=0.0
      SIGZYI(I,J)=0.0
  420 CONTINUE
      PHATZ(J)=0.0
      NNLZ(J)=0
      PHATY(J)=0.0
      NNLY(J)=0
  430 CONTINUE
      FGAIN=2.0 * TWOPI * F3DB / FSAMP
      OD=TWOPI*FSIG/FSAMP
      G0AVG=0.0
      G1AVG=0.0
      G2AVG=0.0
      G3AVG=0.0
      G4AVG=0.0
C  Total samples, samples after settling, number used to compute gain
      NG = 0
      TOTSMP=0.0
      TOTSS=0.0
C
C  Begin seconds loop
C
      DO 7000 KN=1, KMAX
C
C  Begin 10-ms loop
C
      DO 1000 KM = 1, KMOD
C
      CALL INTRPT(LINT)
      IF(LINT.EQ.27) GO TO 9999
      OMEGAT=OD*KM
      IF(KM.EQ.KMOD) OMEGAT=0.0
      COSWAV=COS(OMEGAT)
      SINWAV=SIN(OMEGAT)
      TOTSMP=TOTSMP + 1.0
```

```
C
C  Get six bytes from input file, unpack, and convert to voltages
C
      DO 530 I=1, 6
      CALL GETCHR(LUN,NREC,NLOC,LR1(I),LEND)
      IF(LEND.NE.0) GO TO 85
  530 CONTINUE
      CALL UNPACK(LR2,LR1)
      DO 580 I = 1, 3
      CALL ADVOLT(LR2(I),LR1(I),LGAIN,LCHANO,YW(I))
      YW(I) = YW(I) / CAL(I)
  580 CONTINUE
      CALL ADVOLT(LR2(4),LR1(4),LGAIN,LCHANO,ZW(1))
      ZW(1) = ZW(1) / CAL(4)
C
C  Add signal
C
      ZW(1)=ZW(1) + SR(1)*COSWAV + SI(1)*SINWAV
C
C  Detect without nonlinear processing
      DO 600 I=1,3
      CALL LP(YW(I),YRL(I),YIL(I))
  600 CONTINUE
      DO 605 I=1, ILS
      CALL LP(ZW(I),ZRL(I),ZIL(I))
  605 CONTINUE
C
C  Apply nonlinear processing and obtain narrowband waveforms
C
      DO 610 I=1,3
      CALL NLP(YW(I),PHATY(I),NNLY(I),YR(I),YI(I))
  610 CONTINUE
      DO 620 I=1, ILS
      CALL NLP(ZW(I),PHATZ(I),NNLZ(I),ZR(I),ZI(I))
  620 CONTINUE
C
C  Fixed noise cancellation
C
      ZRF(1) = ZR(1) - YR(IAXIS)
      ZIF(1) = ZI(1) - YI(IAXIS)
C
C  If settled, accumulate measurement statistics
C
      IF(KN.LE.NSET) GO TO 900
      TOTSS=TOTSS +    1
C
C  Bias Zbarl
      CALL MADD(ILS,1,SIGZRL,ZRL,SIGZRL)
      CALL MADD(ILS,1,SIGZIL,ZIL,SIGZIL)
C  Bias Ybarl
      CALL MADD(3,1,SIGYRL,YRL,SIGYRL)
      CALL MADD(3,1,SIGYIL,YIL,SIGYIL)
C  Bias Zbar
      CALL MADD(ILS,1,SIGZR,ZR,SIGZR)
      CALL MADD(ILS,1,SIGZI,ZI,SIGZI)
C  Bias Ybar
      CALL MADD(3,1,SIGYR,YR,SIGYR)
      CALL MADD(3,1,SIGYI,YI,SIGYI)
C  Bias Zbarf
      CALL MADD(ILS,1,SIGZRF,ZRF,SIGZRF)
      CALL MADD(ILS,1,SIGZIF,ZIF,SIGZIF)
C
C  Covariance (z*) * (yT)
      CALL SMULT(ILS,1,ZI,-1.0,CONJ)
      CALL CMVMLT(ILS,1,ILY,ZR,CONJ,YR,YI,DUM1,DUM2,DUM3)
      CALL MADD(ILS,ILY,SIGZYR,DUM1,SIGZYR)
      CALL MADD(ILS,ILY,SIGZYI,DUM2,SIGZYI)
C
C  Covariance (y*) * (yT)
      CALL SMULT(ILY,1,YI,-1.0,CONJ)
      CALL CMVMLT(ILY,1,ILY,YR,CONJ,YR,YI,DUM1,DUM2,DUM3)
```

```
      CALL MADD(ILY,ILY,SIGYYR,DUM1,SIGYYR)
      CALL MADD(ILY,ILY,SIGYYI,DUM2,SIGYYI)
C
C  End 1Ø-ms loop
C
  9ØØ CONTINUE
 1ØØØ CONTINUE
C
C  Zero NLP counters at end of settling period
C
      IF(KN.NE.NSET) GO TO 11ØØ
      DO 1Ø5Ø I=1,3
      NNLZ(I)=Ø
      NNLY(I)=Ø
 1Ø5Ø CONTINUE
 11ØØ CONTINUE
C
C  Time to make estimate?
C
      IF(KN.LE.NSET) GO TO 699Ø
      KSET = KN - NSET
      IF(KSET/KOUT*KOUT.NE.KSET) GO TO 699Ø
C
C  Estimate bias and covariances
C
      CK=1.Ø/TOTSS
C
C  Bias vectors
      CALL SMULT(ILS,1,SIGZRL,CK,ZBARRL)
      CALL SMULT(ILS,1,SIGZIL,CK,ZBARIL)
      CALL SMULT(3,1,SIGYRL,CK,YBARRL)
      CALL SMULT(3,1,SIGYIL,CK,YBARIL)
      CALL SMULT(ILS,1,SIGZRF,CK,ZBARRF)
      CALL SMULT(ILS,1,SIGZIF,CK,ZBARIF)
      CALL SMULT(ILS,1,SIGZR,CK,ZBARR)
      CALL SMULT(ILS,1,SIGZI,CK,ZBARI)
      CALL SMULT(3,1,SIGYR,CK,YBARR)
      CALL SMULT(3,1,SIGYI,CK,YBARI)
C
C  Covariance Ryy
      CALL SMULT(3,3,SIGYYR,CK,RYYR)
      CALL SMULT(3,3,SIGYYI,CK,RYYI)
C
C  Covariance Rzy
      CALL SMULT(ILS,3,SIGZYR,CK,RZYR)
      CALL SMULT(ILS,3,SIGZYI,CK,RZYI)
C
C  Linear estimate
C
      CALL MSE(ILS,SR,SI,ZBARRL,ZBARIL,RMSEL)
C
C  NLP / No-ANC estimate
C
      CALL SMULT(ILS,1,ZBARR,1.Ø,SHATR)
      CALL SMULT(ILS,1,ZBARI,1.Ø,SHATI)
      CALL MSE(ILS,SR,SI,SHATR,SHATI,RMSEN)
C
C  Fixed-cancellation estimate
C
      CALL MSE(ILS,SR,SI,ZBARRF,ZBARIF,RMSEF)
C
C  Invert estimated covariance Ryy of reference
C
      CALL CMI(ILY,ILY2,RYYR,RYYI,RINVR,RINVI,DUM4,DUM5,DUM6,DET)
      IF(DET.NE.Ø.Ø) GO TO 56ØØ
      CALL MZERO(3,3,RINVR)
      CALL MZERO(3,3,RINVI)
 56ØØ CONTINUE
C
C  DMI estimate #Ø
C
```

```
      CALL DMI(ILS,ILY,ILY2,RZYR,RZYI,ZBARR,ZBARI,
     >  YBARR,YBARI,RXYR,RXYI,SHATR,SHATI,WR,WI,DUM3,
     >  DUM4,DUM5,DUM6,RINVR,RINVI)
      CALL MSE(ILS,SR,SI,SHATR,SHATI,RMSEØ)
C
C  DMI estimate #1
C
      CALL DMI(ILS,ILY,ILY2,RZYR,RZYI,ZBARR,ZBARI,
     >  YBARR,YBARI,RXYR,RXYI,SHATR,SHATI,WR,WI,DUM3,
     >  DUM4,DUM5,DUM6,RINVR,RINVI)
      CALL MSE(ILS,SR,SI,SHATR,SHATI,RMSE1)
C
C  DMI estimate #2
C
      CALL DMI(ILS,ILY,ILY2,RZYR,RZYI,ZBARR,ZBARI,
     >  YBARR,YBARI,RXYR,RXYI,SHATR,SHATI,WR,WI,DUM3,
     >  DUM4,DUM5,DUM6,RINVR,RINVI)
      CALL MSE(ILS,SR,SI,SHATR,SHATI,RMSE2)
C
C  Compute improvement factors and output results
C
      T=TOTSS/FSAMP
      GØ=Ø.Ø
      G1=Ø.Ø
      G2=Ø.Ø
      G3=Ø.Ø
      G4=Ø.Ø
      IF(RMSEL.EQ.Ø.Ø) GO TO 676Ø
      IF(RMSEN.EQ.Ø.Ø) GO TO 674Ø
C  Improvement due to clipping
      GØ=2Ø.Ø * ALOG1Ø(RMSEL/RMSEN)
 674Ø IF(RMSE2.EQ.Ø.Ø) GO TO 676Ø
C  Improvements due to ANC
      G1=2Ø.Ø * ALOG1Ø(RMSEN/RMSE2)
C  Total improvement - NLP + ANC
 676Ø G2=GØ + G1
C  Improvement due to fixed cancellation
      IF(RMSEF.EQ.Ø.Ø) GO TO 6765
      G3=2Ø.Ø * ALOG1Ø(RMSEN/RMSEF)
 6765 G4=GØ + G3
      IF(KN.LE.KAVG) GO TO 677Ø
C
C  Average improvements over last half of test
C
      NG=NG+1
      GØAVG=GØAVG + GØ
      G1AVG=G1AVG + G1
      G2AVG=G2AVG + G2
      G3AVG=G3AVG + G3
      G4AVG=G4AVG + G4
 677Ø CONTINUE
      WRITE(3,68ØØ) T, RMSEL, GØ, G1, G2, G3, G4
      WRITE(1Ø,68ØØ) T, RMSEL, GØ, G1, G2, G3, G4
 68ØØ FORMAT(' ',F6.1,G12.4,5F8.1)
C
 699Ø CONTINUE
 7ØØØ CONTINUE
C
C  Output final averages and covariances
C
      GØAVG=GØAVG/NG
      G1AVG=G1AVG/NG
      G2AVG=G2AVG/NG
      G3AVG=G3AVG/NG
      G4AVG=G4AVG/NG
      WRITE(3,79ØØ) GØAVG, G1AVG, G2AVG, G3AVG, G4AVG
      WRITE(1Ø,79ØØ) GØAVG, G1AVG, G2AVG, G3AVG, G4AVG
 79ØØ FORMAT('     AVG',12X,5F8.1)
      WRITE(1Ø,8ØØØ)
 8ØØØ FORMAT(/,' Zbar = ')
      WRITE(1Ø,8Ø1Ø) ZBARR(1), ZBARI(1)
```

```
 8010 FORMAT(' ',G12.4,' +j ',G12.4)
      WRITE(10,8020)
 8020 FORMAT(/,' Ybar = ')
      WRITE(10,8010) (YBARR(I), I=1,3), (YBARI(I), I=1,3)
      WRITE(10,8100)
 8100 FORMAT(/,' Ryy = ')
      DO 8130 I=1,ILY
      CALL CMW(10,3,3,I,RYYR,RYYI)
 8130 CONTINUE
      WRITE(10,8140) DET
 8140 FORMAT(/,' Det(Ryy) = ',G12.4)
      WRITE(10,8200)
 8200 FORMAT(/,' Rxy = ')
      CALL CMW(10,1,3,1,RXYR,RXYI)
      WRITE(10,8300)
 8300 FORMAT(/,' W = ')
      CALL CMW(10,1,3,1,WR,WI)
C
C  End run
C
 9000 CONTINUE
C
C  End disk file
C
 9500 ENDFILE LUN
C
C
 9999 ENDFILE 10
      STOP
      END


      SUBROUTINE DMI(ILS,ILY,ILY2,RZYR,RZYI,ZBARR,ZBARI,YBARR,YBARI,
     >  RXYR,RXYI,SHATR,SHATI,WR,WI,DUM3,DUM4,DUM5,DUM6,RINVR,RINVI)
C  Estimate of signal using DMI ANC algorithm
      DIMENSION RZYR(ILS,ILY), RZYI(ILS,ILY),
     >  ZBARR(ILS), ZBARI(ILS), YBARR(ILY),
     >  YBARI(ILY), SHATR(ILS), SHATI(ILS), WR(ILS,ILY),
     >  WI(ILS,ILY), DUM4(ILY2,ILY2), DUM5(ILY2,ILY2),
     >  DUM6(ILY2,ILY2), RINVR(ILY,ILY), RINVI(ILY,ILY)
C
C  Initial estimate of covariance Rxy (WR and WI are dummies)
C
      CALL SMULT(ILS,1,SHATI,-1.0,SHATI)
      CALL CMVMLT(ILS,1,ILY,SHATR,SHATI,YBARR,YBARI,WR,WI,DUM3)
      CALL SMULT(ILS,ILY,WR,-1.0,WR)
      CALL SMULT(ILS,ILY,WI,-1.0,WI)
      CALL MADD(ILS,ILY,RZYR,WR,RXYR)
      CALL MADD(ILS,ILY,RZYI,WI,RXYI)
C
C  Compute weighting matrix W
C
      CALL CMVMLT(ILS,ILY,ILY,RXYR,RXYI,RINVR,RINVI,WR,WI,DUM3)
C
C  Calculate improvement vector
C
      CALL CMVMLT(ILS,ILY,1,WR,WI,YBARR,YBARI,SHATR,SHATI,DUM3)
C
C  Estimate
C
      CALL SMULT(ILS,1,SHATR,-1.0,SHATR)
      CALL SMULT(ILS,1,SHATI,-1.0,SHATI)
      CALL MADD(ILS,1,SHATR,ZBARR,SHATR)
      CALL MADD(ILS,1,SHATI,ZBARI,SHATI)
      RETURN
      END


      SUBROUTINE MSE(I,SR,SI,ER,EI,RMSE)
C  Mean-square error in estimate of complex vector
      DIMENSION SR(I), SI(I), ER(I), EI(I)
```

```
      RMSE=0.0
      DO 10 J=1,I
      RMSE=RMSE + (SR(J) - ER(J))**2
      RMSE=RMSE + (SI(J) - EI(J))**2
   10 CONTINUE
      RMSE=SQRT(RMSE/I)
      RETURN
      END


      SUBROUTINE NLP(VI,PHAT,NNL,SIHAT,SQHAT)
      COMMON /FLTR/ GAIN, T, TOTSMP, COSWAV, SINWAV
C  Signal-nulling loop with clipper
C
C  Null signal using previous estimates
      VIN=VI - SIHAT*COSWAV - SQHAT*SINWAV
C
C  Average power, find rms, set threshold
      PHAT=PHAT + VIN**2
      RMS=SQRT(PHAT/TOTSMP)
      TAUP= RMS*T
      TAUN=-RMS*T
C
C  Clip input and increment nonlinear event counter
      IF(VIN.LT.TAUP) GO TO 10
      VIN=TAUP
      NNL=NNL+1
      GO TO 100
   10 IF(VIN.GT.TAUN) GO TO 100
      VIN=TAUN
      NNL=NNL+1
      GO TO 100
C
C  Estimate I and Q components of signal
  100 SIHAT=SIHAT + VIN*COSWAV*GAIN
      SQHAT=SQHAT + VIN*SINWAV*GAIN
      RETURN
      END


      SUBROUTINE LP(VI,SIHAT,SQHAT)
      COMMON /FLTR/ GAIN, T, TOTSMP, COSWAV, SINWAV
C  Signal-nulling loop without clipper
C
C  Null signal using previous estimates
      VIN=VI - SIHAT*COSWAV - SQHAT*SINWAV
C
C  Estimate I and Q components of signal
  100 SIHAT=SIHAT + VIN*COSWAV*GAIN
      SQHAT=SQHAT + VIN*SINWAV*GAIN
      RETURN
      END


      SUBROUTINE CMVMLT(I,J,K,A,B,C,D,E,F,DUM)
C  Multiplication of complex matrices
C  E + jF = (A + jB) * (C + jD)
      DIMENSION A(I,J), B(I,J), C(J,K), D(J,K), E(I,K), F(I,K),
     >  DUM(I,K)
C  Real part
      CALL MVMLT(I,J,K,A,C,E)
      CALL MVMLT(I,J,K,B,D,DUM)
      CALL SMULT(I,K,DUM,-1.0,DUM)
      CALL MADD(I,K,E,DUM,E)
C  Imaginary part
      CALL MVMLT(I,J,K,A,D,F)
      CALL MVMLT(I,J,K,B,C,DUM)
      CALL MADD(I,K,F,DUM,F)
      RETURN
      END
```

```
      SUBROUTINE CMI(N,N2,A,B,E,F,G,DUM1,DUM2,DET)
C  Inversion of complex matrix
C  R = A + jB;  Inverse(R) = E + jF
C  Scratch arrays: G, DUM1, DUM2; dimensions N2*N2, N2=N*2
      DIMENSION A(N,N), B(N,N), E(N,N), F(N,N),
     >  G(N2,N2), DUM1(N2,N2), DUM2(N2,N2)
C
C  Load G
C
      DO 2Ø I=1, N
      DO 1Ø J=1, N
      I1=N + I
      J1=N + J
      G(I,J)=A(I,J)
      G(I1,J1)=A(I,J)
      G(I,J1)=B(I,J)
      G(I1,J)=-B(I,J)
   1Ø CONTINUE
   2Ø CONTINUE
C
C  Invert G
C
      CALL INVERT(G,N2,DET,DUM1,DUM2)
C
C  Recover E and F
C
      DO 12Ø I=1,N
      DO 11Ø J=1,N
      J1=J+N
      E(I,J)=G(I,J)
      F(I,J)=G(I,J1)
  11Ø CONTINUE
  12Ø CONTINUE
      RETURN
      END


      SUBROUTINE CMW(LUN,N,M,I,CR,CI)
C  Write out 3-element complex vector or row of matrix
      IMPLICIT LOGICAL (L)
      DIMENSION CR(N,M), CI(N,M)
C
      WRITE(LUN,1Ø) (CR(I,J), J=1,3), (CI(I,J), J=1,3)
   1Ø FORMAT(' [',3G12.4,'] +j [',3G12.4,']')
C
      RETURN
      END


      SUBROUTINE UNPACK(LR2,LR1)
C  Unpack 6 bytes into 8 bytes
      IMPLICIT LOGICAL (L)
      IMPLICIT INTEGER (A-K,M-Z)
      DIMENSION LBIT(8), LR1(1), LR2(1)
      COMMON /ADA/ LBIT, LSR, ABASE, ACBU, ACBL
C
      LR2(1) = LR1(5) .AND. (X'ØF')
      LR2(3) = LR1(6) .AND. (X'ØF')
C
      LR2(2) = Ø
      LR2(4) = Ø
      DO 1Ø I = 1, 4
      J = I + 4
      LTEST = LR1(5) .AND. LBIT(J)
      IF(LTEST.NE.Ø) LR2(2) = LR2(2) .OR. LBIT(I)
      LTEST = LR1(6) .AND. LBIT(J)
      IF(LTEST.NE.Ø) LR2(4) = LR2(4) .OR. LBIT(I)
   1Ø CONTINUE
C
      RETURN
```

```
      END


      SUBROUTINE LIC(L1,L2,I)
C  Converts two bytes to one integer
C  L1 is high-order byte
C  Memory assignment for Microsoft FORTRAN-8Ø
      IMPLICIT LOGICAL (L)
      M1=MLOC$(I)
      CALL POKE(M1,L2)
      M1=M1+1
      CALL POKE(M1,L1)
      RETURN
      END


      SUBROUTINE GETCHR(LUN,NREC,NLOC,LCHR,LEND)
C  Unidirectional single-character read from disk file
      IMPLICIT LOGICAL (L)
      DIMENSION LBUF(128)
C
      IF(NLOC.EQ.1) READ(LUN, REC=NREC, END=1Ø) LBUF
C
      LCHR = LBUF(NLOC)
      LEND = Ø
      NLOC = NLOC + 1
      IF(NLOC.LE.128) RETURN
C
      NLOC = 1
      NREC = NREC + 1
      RETURN
C
   1Ø LCHR = Ø
      LEND = 1
      RETURN
      END


      SUBROUTINE GETIDD(LID,LGAIN,GAIN,NSR,NINT,FREQ)
C  Decode and display identification data in array LID
      IMPLICIT LOGICAL (L)
      DIMENSION LID(128)
C
      WRITE(3,12Ø) (LID(N), N=1, 64)
  12Ø FORMAT(/,' Data identification : ',64A1)
C
      LGAIN = LID(65)
      GAIN = Ø.Ø
      IF(LGAIN.EQ.Ø) GAIN = Ø.5
      IF(LGAIN.EQ.1) GAIN = 2.Ø
      IF(LGAIN.EQ.2) GAIN = 8.Ø
      IF(LGAIN.EQ.3) GAIN = 32.Ø
      WRITE(3,13Ø) GAIN, LGAIN
  13Ø FORMAT(' Gain, gain code = ',F5.2,I5)
C
      CALL LIC(LID(66),LID(67),NSR)
      WRITE(3,14Ø) NSR
  14Ø FORMAT(' Number of 1ØØ-sample sets = ',I8)
C
      CALL LIC(LID(68),LID(69),NINT)
      WRITE(3,15Ø) NINT
  15Ø FORMAT(' Number of 3.3-ms intervals / sample = ',I8)
      IF(NINT.EQ.Ø) GO TO 17Ø
      FREQ = 3ØØ.Ø / NINT
      WRITE(3,16Ø) FREQ
  16Ø FORMAT(' Sampling frequency = ',F1Ø.3)
  17Ø CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE GETCAL(LUN,NREC,NLOC,ND,LGAIN,CAL,NSC,NSS,LERR)
C  Extract calibration constants from noise-data file
      IMPLICIT LOGICAL (L)
      DIMENSION CAL(4), NCAL(4), LR1(6), LR2(4), V(4)
      LERR = Ø
C
C  First disk of set?
C
      IF(ND.EQ.1) GO TO 5
      NSC = Ø
      GO TO 62Ø
C
C  Do data contain calibration signal?
C
    5 WRITE(3,1Ø)
   1Ø FORMAT(/,' Calibration signal included in data ? ')
      READ(3,2Ø) LCAL
   2Ø FORMAT(A1)
      LCAL=LUPPER(LCAL)
      IF(LCAL.EQ.89) GO TO 2ØØ
C
C  Calibration signal not present, get values from operator
C
      NSC=Ø
      WRITE(3,11Ø)
  11Ø FORMAT(' Level of 1-V calibration signals for',/,
     > ' Channels 1, 2, 3, 4 = ')
      DO 13Ø N = 1, 4
      CALL ATR(CAL(N),IERR)
  13Ø CONTINUE
      GO TO 6ØØ
C
C  Calibration signal present, find values
C
C  Get duration and approximate value
  2ØØ CONTINUE
      WRITE(3,21Ø)
  21Ø FORMAT(' Duration of calibration signal in seconds = ')
      CALL ATI(NSC,IERR)
      WRITE(3,22Ø)
  22Ø FORMAT(' Approximate calibration-signal voltage = ')
      CALL ATR(CALAPX,IERR)
C
C  Set window and zero averages and counters
      W1 = Ø.9 * CALAPX
      W2 = 1.1 * CALAPX
      DO  23Ø N = 1, 4
      NCAL(N) = Ø
      CAL(N) = Ø.Ø
  23Ø CONTINUE
C
C  Process calibration signal
C
      DO 4ØØ NS = 1, NSC
      DO 39Ø N1 = 1, 1ØØ
C
C  Get and unpack data set
C
C  Get six bytes from disk
      DO 3ØØ N=1, 6
      CALL GETCHR(LUN,NREC,NLOC,LR1(N),LEND)
      IF(LEND.NE.Ø) GO TO 9ØØ
  3ØØ CONTINUE
C
C  Unpack and print
      CALL UNPACK(LR2,LR1)
      DO 31Ø N=1, 4
      CALL ADVOLT(LR2(N),LR1(N),LGAIN,LCHD,V(N))
  31Ø CONTINUE
C
```

```
C  Average if within window
       DO 37Ø N = 1, 4
       V(N) = ABS(V(N))
       IF((V(N).LT.W1) .OR. (V(N).GT.W2)) GO TO 36Ø
       CAL(N) = CAL(N) + V(N)
       NCAL(N) = NCAL(N) + 1
   36Ø CONTINUE
   37Ø CONTINUE
C
   39Ø CONTINUE
   4ØØ CONTINUE
C
C  Scale averages
C
       DO 42Ø N = 1, 4
       IF(NCAL(N).LE.Ø) GO TO 8ØØ
       CAL(N) = CAL(N) / NCAL(N)
   42Ø CONTINUE
       WRITE(3,43Ø) (CAL(N), N=1,4)
   43Ø FORMAT(' 1-V Calibration levels = ',4F9.3)
C
C  Get start time
C
   6ØØ CONTINUE
       WRITE(3,61Ø)
   61Ø FORMAT(' Start time = ')
       CALL ATI(NSS,IERR)
C
C  Skip to start
C
   62Ø NSKIP = NSS - NSC
       IF(NSKIP.EQ.Ø) RETURN
       DO 65Ø NS = 1, NSKIP
       DO 64Ø N1 = 1, 1ØØ
       DO 63Ø N=1, 6
       CALL GETCHR(LUN,NREC,NLOC,LR1(N),LEND)
       IF(LEND.NE.Ø) GO TO 9ØØ
   63Ø CONTINUE
   64Ø CONTINUE
   65Ø CONTINUE
C
C  Done - normal exit
C
       RETURN
C
C  Error
C
   8ØØ WRITE(3,81Ø) N
   81Ø FORMAT(/,' *** NO CALIBRATION SIGNAL ON CHANNEL ',I5)
   9ØØ LERR=1
       RETURN
       END
```

APPENDIX D. GAIN STATISTICS

NUMERICAL DATA - AVERAGES

Single-channel NLP/ANC algorithm with disk-file input
Program # FHR-353E

Noise-file parameters:

A/D gain = 2.0 Sampling frequency = 100.0
Total seconds in noise file = 575

Processing parameters:

Signal = .1414 +j 0.000
Nsec = 270 Kout = 2 Nrun = 2
Nset = 2 F3dB = 1.00 Thresh = infinite

Set	Axis	Date	Time	Comment	NSC	NSS	Gn	Ga	Gta	Gf	Gtf
1-1	Y	01/14	16:45	GWX, SD	10	17	0.0	9.5	9.5	8.4	8.4
1-1							0.0	10.4	10.4	11.1	11.1
1-2					0	17	0.0	7.8	7.8	11.7	11.7
1-2							0.0	10.7	10.7	3.9	3.9
1-3					0	17	0.0	16.3	16.3	15.1	15.1
1-3							0.0	1.3	1.3	2.1	2.1
2-1	Z	01/14	18:00	GWX	10	12	0.0	2.9	2.9	1.9	1.9
2-1							0.0	.9	.9	-2.2	-2.2
2-2					0	12	0.0	-.8	-.8	-1.1	-1.1
2-2							0.0	-.8	-.8	-1.1	-1.1
2-3					0	12	0.0	-3.9	-3.9	-3.8	-3.8
2-3							0.0	-3.6	-3.6	2.9	2.9
3-1	Y	01/05	16:00		24	28	0.0	19.7	19.7	14.0	14.0
3-1							0.0	11.6	11.6	9.8	9.8
3-2					0	28	0.0	1.5	1.5	1.6	1.6
3-2							0.0	6.3	6.3	8.8	8.8
3-3					0	28	0.0	11.9	11.9	14.8	14.8
3-4							0.0	7.5	7.5	3.7	3.7
4-1	X	01/11	05:30	GWX	11	13	0.0	7.0	7.0	11.5	11.5
4-1							0.0	6.0	6.0	8.4	8.4
4-2					0	13	0.0	4.4	4.4	5.6	5.6
4-2							0.0	5.2	5.2	6.7	6.7
4-3					0	13	0.0	3.8	3.8	1.6	1.6
4-3							0.0	2.4	2.4	2.4	2.4
5-1	Y	01/11	06:20	GWX	16	19	0.0	7.2	7.2	4.9	4.9
5-1							0.0	26.2	26.2	14.3	14.3
5-2					0	19	0.0	14.3	14.3	5.6	5.6
5-2							0.0	7.0	7.0	12.3	12.3
5-3					0	19	0.0	4.2	4.2	4.8	4.8
5-3							0.0	6.4	6.4	4.9	4.9
6-1	Z	01/05	18:00		12	22	0.0	-1.3	-1.3	.1	.1
6-1							0.0	-4.5	-4.5	-2.1	-2.1
6-2					0	22	0.0	.4	.4	-.5	-.5
6-2							0.0	-2.2	-2.2	-1.5	-1.5
6-3					0	22	0.0	-.7	-.7	-3.4	-3.4
6-3							0.0	.3	.3	5.3	5.3
7-1	Z	01/11	07:30	GWX	9	12	0.0	2.4	2.4	1.9	1.9
7-1							0.0	-6.7	-6.7	-6.4	-6.4
7-2					0	12	0.0	-.9	-.9	1.3	1.3
7-2							0.0	-.5	-.5	.1	.1
7-3					0	12	0.0	-3.4	-3.4	-.3	-.3
7-3							0.0	-1.6	-1.6	-1.5	-1.5

8-1	X	Ø1/Ø7	Ø9:4Ø	GWX	11	18	Ø.Ø	6.4	6.4	7.1	7.1
8-1							Ø.Ø	12.6	12.6	12.7	12.7
8-2					Ø	18	Ø.Ø	2.5	2.5	6.5	6.5
8-2							Ø.Ø	-.9	-.9	.2	.2
8-3					Ø	18	Ø.Ø	9.1	9.1	13.9	13.9
8-3							Ø.Ø	1Ø.8	1Ø.8	1Ø.2	1Ø.2
9-1	X	Ø1/Ø5	11:1Ø	GWX 528s	4Ø	45	Ø.Ø	9.4	9.4	11.7	11.7
9-1							Ø.Ø	9.8	9.8	7.7	7.7
9-2					Ø	1Ø	Ø.Ø	7.9	7.9	5.7	5.7
9-2							Ø.Ø	1.8	1.8	6.3	6.3
9-3					Ø	1Ø	Ø.Ø	7.9	7.9	5.7	5.7
9-3							Ø.Ø	9.5	9.5	8.2	8.2
1Ø-1	Z	Ø1/Ø7	11:2Ø	GWX	9	11	Ø.Ø	-3.5	-3.5	-2.7	-2.7
1Ø-1							Ø.Ø	-2.4	-2.4	-2.4	-2.4
1Ø-2					Ø	1Ø	Ø.Ø	-4.2	-4.2	-1.3	-1.3
1Ø-2							Ø.Ø	-4.Ø	-4.Ø	-2.8	-2.8
1Ø-3					Ø	1Ø	Ø.Ø	-4.5	-4.5	-1.1	-1.1
1Ø-3							Ø.Ø	-3.4	-3.4	1.1	1.1
11-1	X	Ø1/Ø6	19:15		8	1Ø	Ø.Ø	2.1	2.1	-9.7	-9.7
11-1							Ø.Ø	1.8	1.8	-.5	-.5
11-2					Ø	1Ø	Ø.Ø	-6.Ø	-6.Ø	-5.Ø	-5.Ø
11-2							Ø.Ø	-6.5	-6.5	-15.Ø	-15.Ø
11-3					Ø	1Ø	Ø.Ø	-.6	-.6	-6.6	-6.6
11-3							Ø.Ø	-3.2	-3.2	-9.8	-9.8
12-1	Y	Ø1/12	19:3Ø	GWX	3	4	Ø.Ø	16.3	16.3	19.8	19.8
12-1							Ø.Ø	11.2	11.2	8.5	8.5
12-2					Ø	4	Ø.Ø	18.7	18.7	22.6	22.6
12-2							Ø.Ø	18.4	18.4	15.7	15.7
12-3					Ø	4	Ø.Ø	18.5	18.5	12.1	12.1
12-3							Ø.Ø	11.6	11.6	8.8	8.8
13-1	X	Ø1/14	14:ØØ	Disturbed	3	6	Ø.Ø	-2.4	-2.4	-3.8	-3.8
13-1							Ø.Ø	.3	.3	-.1	-.1
13-2					Ø	6	Ø.Ø	4.Ø	4.Ø	4.Ø	4.Ø
13-2							Ø.Ø	4.4	4.4	4.7	4.7
13-3					Ø	6	Ø.Ø	-3.Ø	-3.Ø	-3.6	-3.6
13-3							Ø.Ø	-1.7	-1.7	-6.4	-6.4
14-1	Y	Ø1/Ø7	1Ø:3Ø	GWX	1Ø	14	Ø.Ø	9.1	9.1	12.8	12.8
14-1							Ø.Ø	4.6	4.6	1Ø.4	1Ø.4
14-2					Ø	14	Ø.Ø	13.7	13.7	11.4	11.4
14-2							Ø.Ø	13.3	13.3	6.2	6.2
14-3					Ø	14	Ø.Ø	5.8	5.8	9.8	9.8
14-3							Ø.Ø	6.9	6.9	1.4	1.4

NUMERICAL DATA - AVERAGES

Single-channel NLP/ANC algorithm with disk-file input
Program # FHR-353E

Noise-file parameters:

A/D gain = 2.Ø Sampling frequency = 1ØØ.Ø
Total seconds in noise file = 575

Processing parameters:

Signal = .1414 +j Ø.ØØØ
Nsec = 27Ø Kout = 2 Nrun = 2
Nset = 2 F3dB = 1.ØØ Thresh = 2.828

Set	Axis	Date	Time	Comment	NSC	NSS	Gn	Ga	Gta	Gf	Gtf
1-1	Y	Ø1/14	16:45	GWX, SD	1Ø	17	.4	9.5	9.9	9.6	1Ø.Ø
1-1							-.4	3.2	2.7	3.1	2.6
1-2					Ø	17	-2.5	3.8	1.3	5.4	2.9

1-2								2.0	6.8	8.8	2.1	4.1
1-3						0	17	1.4	23.0	24.4	10.3	11.7
1-3								-2.8	3.3	.5	4.2	1.5
2-1	Z	01/14	18:00	GWX		10	12	-6.4	3.3	-3.1	1.0	-5.4
2-1								-3.9	1.4	-2.5	-1.4	-5.3
2-2						0	12	-2.9	-.5	-3.4	-1.5	-4.4
2-2								4.4	-1.2	3.2	-2.3	2.1
2-3						0	12	-3.1	-3.0	-6.1	-3.1	-6.1
2-3								-1.1	-2.9	-4.0	-1.5	-2.6
3-1	Y	01/05	16:00			24	28	.6	20.3	20.9	14.3	14.9
3-1								.9	10.4	11.3	8.4	9.3
3-2						0	28	.8	-1.5	-.6	-.0	.8
3-2								.4	5.4	5.8	7.5	7.9
3-3						0	28	.0	13.5	13.5	15.9	15.9
3-3								.2	7.1	7.2	3.7	3.9
4-1	X	01/11	05:30	GWX		11	13	.2	8.1	8.3	7.8	8.0
4-1								-.2	5.6	5.4	7.2	7.0
4-2						0	13	-.1	4.1	4.1	5.4	5.3
4-2								-.6	4.9	4.3	7.1	6.5
4-3						0	13	.3	3.3	3.6	.9	1.3
4-3								.3	2.4	2.8	2.7	3.0
5-1	Y	01/11	06:20	GWX		16	19	.6	7.2	7.8	4.9	5.5
5-1								-.1	25.3	25.2	12.8	12.7
5-2						0	19	1.2	12.9	14.1	5.0	6.3
5-2								-.5	5.7	5.3	11.9	11.4
5-3						0	19	-.1	3.7	3.6	5.0	4.9
5-3								-1.2	7.6	6.4	7.1	5.9
6-1	Z	01/05	18:00			12	22	-5.7	-.2	-5.9	-.0	-5.7
6-1								1.4	.2	1.7	1.8	3.2
6-2						0	22	1.0	.6	1.7	-.2	.8
6-2								1.0	-2.2	-1.3	-2.0	-1.0
6-3						0	22	-1.0	-.9	-2.0	-3.0	-4.1
6-3								-4.0	.3	-3.7	3.2	-.8
7-1	Z	01/11	07:30	GWX		9	12	.0	2.7	2.7	1.8	1.9
7-1								-.1	-5.1	-5.2	-4.4	-4.5
7-2						0	12	-6.4	-.6	-7.0	.8	-5.6
7-2								-.8	-.5	-1.3	.6	-.2
7-3						0	12	-1.0	-3.6	-4.6	-.5	-1.5
7-3								-1.4	-.9	-2.3	-3.7	-5.1
8-1	X	01/07	09:40	GWX		11	18	1.0	7.6	8.6	9.7	10.7
8-1								.1	12.9	13.0	12.3	12.4
8-2						0	18	-.8	1.5	.7	5.5	4.6
8-2								0.0	-.9	-.9	.3	.3
8-3						0	18	-.2	8.4	8.2	12.5	12.4
8-3								.1	10.3	10.4	9.8	9.8
9-1	X	01/05	11:10	GWX	528s	40	45	.1	9.8	9.9	12.6	12.7
9-1								-.3	8.8	8.5	7.1	6.8
9-2						0	10	.3	8.2	8.5	6.0	6.3
9-2								1.6	.8	2.4	6.0	7.6
9-3						0	10	.4	12.6	13.1	14.7	15.1
9-3								.7	9.1	9.7	7.4	8.1
10-1	Z	01/07	11:20	GWX		9	11	-.7	-3.2	-3.9	-3.0	-3.7
10-1								.8	-2.5	-1.7	-2.2	-1.4
10-2						0	10	-.5	-4.0	-4.5	-1.8	-2.2
10-2								1.2	-4.1	-3.0	-2.7	-1.6
10-3						0	10	.2	-4.9	-4.7	-.6	-.4
10-3								.8	-3.6	-2.7	.7	1.5
11-1	X	01/06	19:15			8	10	.8	-.0	.8	-9.5	-8.7
11-1								-.6	-2.2	-2.8	-2.1	-2.7
11-2						0	10	.6	-5.5	-4.8	-5.4	-4.8
11-2								-1.6	-6.6	-8.2	-14.3	-15.9

11-3					Ø	1Ø	-.4	-.7	-1.Ø	-6.8	-7.2
11-3							.2	-2.9	-2.7	-1Ø.3	-1Ø.1
12-1	Y	Ø1/12	19:3Ø	GWX	3	4	.4	15.1	15.5	-3.8	-3.4
12-1							-.3	11.7	11.4	-1.5	-1.8
12-2					Ø	4	-.4	22.8	22.4	-7.1	-7.5
12-2							.1	12.8	12.9	-2.8	-2.7
12-3					Ø	4	-.1	19.5	19.4	12.1	12.Ø
12-3							-.1	9.9	9.8	7.4	7.3
13-1	X	Ø1/14	14:ØØ	Disturbed	3	6	Ø.Ø	-2.6	-2.6	-3.9	-3.9
13-1							.Ø	.2	.2	-.2	-.2
13-2					Ø	6	Ø.Ø	4.1	4.1	4.4	4.4
13-2							Ø.Ø	5.3	5.3	5.3	5.3
13-3					Ø	6	Ø.Ø	-2.8	-2.8	-3.5	-3.5
13-3							-.1	-1.7	-1.8	-6.3	-6.4
14-1	Y	Ø1/Ø7	1Ø:3Ø	GWX	1Ø	14	.4	7.4	7.8	11.7	12.1
14-1							-.6	4.3	3.7	9.3	8.7
14-2					Ø	14	.2	12.9	13.1	1Ø.8	11.Ø
14-2							-.Ø	12.3	12.3	5.5	5.4
14-3					Ø	14	.Ø	6.6	6.6	11.6	11.6
14-3							6.3	-5.2	1.1	-7.7	-1.4

NUMERICAL DATA - AVERAGES

Single-channel NLP/ANC algorithm with disk-file input
Program # FHR-353E

Noise-file parameters:

A/D gain = 2.Ø Sampling frequency = 1ØØ.Ø
Total seconds in noise file = 575

Processing parameters:

Signal = .1414 +j Ø.ØØØ
Nsec = 27Ø Kout = 2 Nrun = 2
Nset = 2 F3dB = 1.ØØ Thresh = 1.414

Set	Axis	Date	Time	Comment	NSC	NSS	Gn	Ga	Gta	Gf	Gtf
1-1	Y	Ø1/14	16:45	GWX, SD	1Ø	17	4.8	4.1	8.9	-.3	4.5
1-1							.8	1.Ø	1.8	.Ø	.8
1-2					Ø	17	-2.4	3.1	.7	4.2	1.7
1-2							1.9	1Ø.2	12.1	4.1	5.9
1-3					Ø	17	4.8	4.1	8.9	-.3	4.5
1-3							.8	1.Ø	1.8	.Ø	.8
2-1	Z	Ø1/14	18:ØØ	GWX	1Ø	12	-9.2	3.6	-5.5	1.2	-7.9
2-1							-3.9	1.8	-2.1	-2.3	-6.1
2-2					Ø	12	-.9	-.6	-1.5	-.1	-1.Ø
2-2							3.2	3.8	7.Ø	-1.Ø	2.2
2-3					Ø	12	-1.3	-3.8	-5.1	-4.4	-5.7
2-3							2.6	-3.4	-.7	-5.1	-2.5
3-1	Y	Ø1/Ø5	16:ØØ		24	28	.5	15.5	16.Ø	14.5	14.9
3-1							2.Ø	14.7	16.7	13.2	15.2
3-2					Ø	28	2.2	-3.5	-1.3	-2.9	-.7
3-2							-.7	1Ø.3	9.6	2Ø.4	19.7
3-3					Ø	28	.1	8.3	8.3	1Ø.2	1Ø.3
3-3							-3.5	8.7	5.2	5.8	2.3
4-1	X	Ø1/11	Ø5:3Ø	GWX	11	13	-3.6	5.6	2.1	1Ø.3	6.7
4-1							.2	3.4	3.6	9.7	9.9
4-2					Ø	13	-.2	7.8	7.6	1Ø.8	1Ø.6
4-2							-.1	4.Ø	3.8	5.3	5.2
4-3					Ø	13	-.3	3.8	3.4	2.5	2.1
4-3							-.6	5.1	4.5	6.6	6.Ø

5-1	Y	01/11	06:20	GWX	16	19	4.5	13.7	18.2	14.9	19.4
5-1							1.6	18.2	19.8	11.8	13.4
5-2					0	19	9.0	-6.0	3.0	-8.9	.1
5-2							-1.3	11.1	9.8	12.0	10.7
5-3					0	19	-.7	8.3	7.6	9.1	8.4
5-3							-.5	12.1	11.5	11.7	11.2
6-1	Z	01/05	18:00		12	22	-7.1	-.6	-7.7	.9	-6.3
6-1							-2.0	1.1	-.9	-2.9	-4.9
6-2					0	22	1.5	-.0	1.4	3.5	4.9
6-2							4.5	-3.3	1.2	-3.9	.5
6-3					0	22	8.1	-4.0	4.1	-7.0	1.1
6-3							-9.1	.3	-8.8	2.5	-6.7
7-1	Z	01/11	07:30	GWX	9	12	.1	1.8	1.9	3.8	3.9
7-1							-3.0	-4.4	-7.4	-5.4	-8.4
7-2					0	12	-9.2	-.5	-9.7	.3	-8.9
7-2							-.3	-1.1	-1.3	1.1	.9
7-3					0	12	-2.3	-3.7	-5.9	-1.8	-4.0
7-3							-1.4	-2.7	-4.1	-2.5	-4.0
8-1	X	01/07	09:40	GWX	11	18	.2	6.9	7.2	8.4	8.6
8-1							1.0	11.8	12.9	11.4	12.4
8-2					0	18	-3.5	8.7	5.3	10.3	6.8
8-2							-1.3	-.8	-2.1	.6	-.7
8-3					0	18	.9	12.2	13.1	6.4	7.2
8-3							3.4	-2.6	.8	-5.1	-1.7
9-1	X	01/05	11:10	GWX 528s	40	45	1.4	19.5	20.9	21.5	22.9
9-1							-.7	4.3	3.6	3.7	3.0
9-2					0	10	2.2	7.5	9.7	6.6	8.8
9-2							1.6	-1.0	.6	2.9	4.5
9-3					0	10	1.7	5.0	6.7	9.9	11.6
9-3							2.6	9.1	11.8	4.3	6.9
10-1	Z	01/07	11:20	GWX	9	11	1.3	-3.8	-2.5	-6.3	-5.0
10-1							.5	-2.6	-2.1	-.9	-.4
10-2					0	10	-1.2	-3.7	-5.0	-1.5	-2.8
10-2							2.1	-3.6	-1.5	-1.0	1.1
10-3					0	10	1.3	-5.3	-4.1	.3	1.6
10-3							.2	-3.3	-3.0	.9	1.2
11-1	X	01/06	19:15		8	10	-1.4	4.5	3.1	-6.6	-8.0
11-1							7.8	-6.8	1.1	-11.3	-3.4
11-2					0	10	6.0	-8.2	-2.2	-11.5	-5.5
11-2							-6.4	2.9	-3.6	-11.2	-17.6
11-3					0	10	-1.7	.1	-1.6	-7.5	-9.2
11-3							-.2	-3.2	-3.3	-11.6	-11.7
12-1	Y	01/12	19:30	GWX	3	4	1.5	6.3	7.8	6.7	8.2
12-1							-.4	10.5	10.1	7.2	6.8
12-2					0	4	-.5	19.1	18.6	17.2	16.7
12-2							.3	4.7	5.0	6.2	6.5
12-3					0	4	.7	15.9	16.5	9.1	9.8
12-3							1.1	12.7	13.8	8.9	10.0
13-1	X	01/14	14:00	Disturbed	3	6	.6	.5	1.1	-1.2	-.6
13-1							1.1	.8	1.9	.4	1.5
13-2					0	6	-3.6	2.7	-.8	2.4	-1.1
13-2							5.9	3.0	8.9	2.8	8.6
13-3					0	6	-4.5	-.8	-5.3	-1.1	-5.6
13-3							-2.8	-1.3	-4.0	-5.2	-8.0
14-1	Y	01/07	10:30	GWX	10	14	2.8	6.9	9.6	4.7	7.5
14-1							-1.1	5.1	4.1	4.1	3.0
14-2					0	14	-.7	7.2	6.5	7.0	6.3
14-2							-.7	14.4	13.7	7.6	6.9
14-3					0	14	-1.0	10.6	9.6	12.3	11.3
14-3							1.4	-8.8	-7.3	-8.5	-7.1

NUMERICAL DATA - AVERAGES

Single-channel NLP/ANC algorithm with disk-file input
Program # FHR-353E

Noise-file parameters:

A/D gain = 2.0 Sampling frequency = 100.0
Total seconds in noise file = 575

Processing parameters:

Signal = .1414 +j 0.000
Nsec = 270 Kout = 2 Nrun = 2
Nset = 2 F3dB = 1.00 Thresh = 0.707

Set	Axis	Date	Time	Comment	NSC	NSS	Gn	Ga	Gta	Gf	Gtf
1-1	Y	01/14	16:45	GWX, SD	10	17	-3.1	6.1	3.0	8.7	5.6
1-1							-.4	-1.3	-1.8	-1.9	-2.3
1-2					0	17	-1.9	5.6	3.8	8.9	7.1
1-2							5.0	7.4	12.4	4.6	9.6
1-3					0	17	11.5	-8.2	3.3	-10.6	.9
1-3							2.8	-1.4	1.4	-2.3	.6
2-1	Z	01/14	18:00	GWX	10	12	-8.7	3.1	-5.7	-1.6	-10.3
2-1							-5.1	.7	-4.4	-3.2	-8.3
2-2					0	12	-1.7	-1.2	-2.9	-2.6	-4.3
2-2							2.9	.9	3.8	-3.5	-.6
2-3					0	12	-1.2	-3.4	-4.5	-6.8	-8.0
2-3							9.3	-6.2	3.1	-11.8	-2.5
3-1	Y	01/05	16:00		24	28	-1.1	13.3	12.2	15.5	14.4
3-1							4.1	17.8	21.9	10.2	14.3
3-2					0	28	-2.6	-1.9	-4.6	2.1	-.5
3-2							-1.1	20.3	19.2	4.4	3.3
3-3					0	28	2.5	3.2	5.7	4.1	6.7
3-4							-7.2	5.2	-2.0	4.9	-2.3
4-1	X	01/11	05:30	GWX	11	13	-4.9	3.3	-1.6	6.7	1.7
4-1							3.2	-1.5	1.6	5.7	8.9
4-2					0	13	.3	11.6	11.9	11.5	11.8
4-2							1.0	6.1	7.1	6.6	7.6
4-3					0	13	-.0	-.1	-.2	-.8	-.8
4-3							-2.2	5.8	3.6	8.6	6.4
5-1	Y	01/11	06:20	GWX	16	19	7.5	-1.6	5.9	-5.2	2.3
5-1							1.7	13.0	14.7	14.2	15.9
5-2					0	19	7.0	-8.0	-1.0	-10.6	-3.7
5-2							1.5	2.5	4.0	.5	2.0
5-3					0	19	3.7	.1	3.7	-2.5	1.2
5-3							.9	7.6	8.5	8.2	9.1
6-1	Z	01/05	18:00		12	22	-10.3	-.4	-10.7	.7	-9.6
6-1							-5.2	1.9	-3.3	-4.8	-10.1
6-2					0	22	-.8	1.7	.9	-2.0	-2.8
6-2							6.9	-3.6	3.3	-4.0	3.0
6-3					0	22	5.6	.3	5.9	-5.0	.6
6-3							-9.3	.8	-8.5	4.0	-5.3
7-1	Z	01/11	07:30	GWX	9	12	-1.0	-.0	-1.1	6.4	5.4
7-1							-2.1	-4.8	-6.9	-5.4	-7.5
7-2					0	12	-7.3	-.5	-7.8	1.0	-6.3
7-2							4.1	-4.3	-.2	.0	4.1
7-3					0	12	-3.3	-3.8	-7.0	-3.9	-7.1
7-3							-.4	-3.4	-3.8	-4.7	-5.1
8-1	X	01/07	09:40	GWX	11	18	-.3	4.1	3.8	5.7	5.4
8-1							2.2	10.9	13.0	10.5	12.7
8-2					0	18	-4.6	7.0	2.4	6.3	1.7
8-2							-4.0	-.6	-4.6	.5	-3.5

8-3					0	18	2.7	9.2	11.9	1.5	4.2
8-3							1.5	-4.2	-2.7	-6.8	-5.3
9-1	X	01/05	11:10	GWX 528s	40	45	3.7	15.8	19.5	11.1	14.7
9-1							-.2	2.5	2.3	2.4	2.2
9-2					0	10	3.8	2.7	6.5	1.5	5.3
9-2							1.7	-3.5	-1.8	-.7	1.0
9-3					0	10	4.9	-1.4	3.5	.0	4.9
9-3							2.9	7.6	10.5	3.0	5.9
10-1	Z	01/07	11:20	GWX	9	11	6.1	-2.5	3.6	-12.9	-6.8
10-1							.9	-2.2	-1.4	-2.0	-1.1
10-2					0	10	-1.7	-2.8	-4.5	-1.6	-3.3
10-2							-1.2	-2.9	-4.1	-.5	-1.6
10-3					0	10	1.0	-3.6	-2.7	3.2	4.1
10-3							-.2	-3.2	-3.4	.2	-.1
11-1	X	01/06	19:15		8	10	-3.9	3.2	-.6	-6.9	-10.8
11-1							6.1	-8.5	-2.4	-14.5	-8.4
11-2					0	10	-4.2	-1.6	-5.8	-4.2	-8.4
11-2							-10.3	3.4	-6.9	-10.9	-21.2
11-3					0	10	-2.8	-1.2	-4.0	-7.3	-10.1
11-3							-5.7	-.4	-6.1	-5.6	-11.3
12-1	Y	01/12	19:30	GWX	3	4	-.5	5.4	4.9	7.5	7.1
12-1							-.0	9.9	9.8	6.3	6.2
12-2					0	4	-2.8	8.3	5.5	8.1	5.3
12-2							.2	3.2	3.4	3.9	4.2
12-3					0	4	.9	11.9	12.8	7.8	8.8
12-3							4.5	7.8	12.3	4.6	9.0
13-1	X	01/14	14:00	Disturbed	3	6	-5.4	2.0	-3.4	1.4	-4.0
13-1							5.4	1.1	6.5	2.9	8.3
13-2					0	6	-8.7	1.0	-7.7	1.6	-7.0
13-2							.3	-.5	-.2	2.3	2.5
13-3					0	6	-7.4	-.2	-7.6	.1	-7.3
13-3							-8.7	-.6	-9.3	-6.9	-15.6
14-1	Y	01/07	10:30	GWX	10	14	8.8	3.1	11.9	-3.5	5.3
14-1							-.4	9.8	9.3	3.4	3.0
14-2					0	14	8.8	3.1	11.9	-3.5	5.3
14-2							-.4	9.8	9.3	3.4	3.0
14-3					0	14	-2.0	4.8	2.8	4.7	2.7
14-3							-1.7	13.0	11.3	14.7	13.1